Refine Search

Search Results -

Terms	Documents
L16 or L15 or L14	18

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L17

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Saturday, November 10, 2007 **Purge Queries** Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L17	L16 or L15 or L14	18	L17
L16	Firmware ADJ volume	7	L16
L15	PEIMs	4	L15
L14	PRE-EFI	10	L14
L13	L12 and (parameters or signature or (call adj structure) or (call adj graph) or stamping or stamp or coupling or arguments or metadata or (meta adj data))	33	L13
L12	L11 and ((DAG) Or (directed ADJ Acyclic ADJ Graph))	60	L12
L11	717/157 |717/159 |717/133 |717/144 |717/151 |717/156 |717/132 |717/153 |717/140.ccls or 714/38.ccls. or (712/241).ccls.	2762	L11
L10	L9 not L6	3	L10
L9	L5 and (parameters or signature or (call adj structure) or (call adj graph) or stamping or stamp or coupling or arguments or metadata or (meta adj data))	253	L9
L8	L5 and (optimize or optimization).ab.	3	L8

L7	L6 and EFI	2	L7
L6	L5 and (parameters or signature or (call adj structure) or (call adj graph) or stamping or stamp or coupling or arguments)	250	L6
L5	L4 and 31	437	L5
L4	firmware.ab.	1098	L4
L3	((DAG) Or (directed ADJ Acyclic ADJ Graph))	2593	L3
L2	L1 and ((DAG) Or (directed ADJ Acyclic ADJ Graph))	32	L2
L1	(717/157 |717/159 |717/133 |717/144).ccls.	502	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L23 and call	28

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L24

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Saturday, November 10, 2007 **Purge Queries** Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L24	L23 and call	28	L24
L23	((DAG) Or (directed ADJ Acyclic ADJ Graph)).ab.	89	L23
L22	L21 AND call	39	L22
L21	L20 and (order or reorder)	116	L21
L20	L19 and profile	119	L20
L19	L18 and optimization	542	L19
L18	((DAG) Or (directed ADJ Acyclic ADJ Graph))	2593	L18
L17	L16 or L15 or L14	18	L17
L16	Firmware ADJ volume	7	L16
L15	PEIMs	4	L15
L14	PRE-EFI	10	L14
L13	L12 and (parameters or signature or (call adj structure) or (call adj graph) or stamping or stamp or coupling or arguments or metadata or (meta adj data))	33	L13

L12	L11 and ((DAG) Or (directed ADJ Acyclic ADJ Graph))	60	L12
L11	717/157 |717/159 |717/133 |717/144 |717/151 |717/156 |717/132 |717/153 |717/140.ccls or 714/38.ccls. or (712/241).ccls.	2762	L11
L10	L9 not L6	3	L10
L9	L5 and (parameters or signature or (call adj structure) or (call adj graph) or stamping or stamp or coupling or arguments or metadata or (meta adj data))	253	L9
L8	L5 and (optimize or optimization).ab.	3	L8
L7	L6 and EFI	2	L7
L6	L5 and (parameters or signature or (call adj structure) or (call adj graph) or stamping or stamp or coupling or arguments)	250	L6
L5	L4 and 31	437	L5
L4	firmware.ab.	1098	L4
L3	((DAG) Or (directed ADJ Acyclic ADJ Graph))	2593	L3
L2	L1 and ((DAG) Or (directed ADJ Acyclic ADJ Graph))	32	L2
L1	(717/157 |717/159 |717/133 |717/144).ccls.	502	L1

END OF SEARCH HISTORY

```
File 348:EUROPEAN PATENTS 1978-2007/ 200723
         (c) 2007 European Patent Office
File 349:PCT FULLTEXT 1979-2007/UB=20070607UT=20070531
         (c) 2007 WIPO/Thomson

Set     Items   Description
S1     219417   DAG OR DIGRAPH OR GRAPH? ? OR DEPENDENCY OR CALL()(DIAGRAM?
              ? OR GRAPH? ? OR CHART? ?) OR TOPOLOGICAL()ORDER???
S2     158745   (CALL??? OR INVOK??? OR INVOCATION)(7N)(PROGRAM? ? OR OBJE-
             CT? ? OR CODE OR CLASS OR CLASSES OR MODULE? ? OR APPLICATION?
              ? OR SOFTWARE OR FIRMWARE OR FILE? ? OR METHOD? ? OR ROUTINE?
              ? OR SUBROUTINE? ? OR THREAD? ? OR PROCESS OR PROCESSES)
S3     159469   (PASS OR PASSES OR PASSING OR PASSAGE OR PASSED OR STAMP???
             OR COUPL??? OR RECEIV???)(5N)(MESSAGE? ? OR ARGUMENT? ? OR P-
             ARAMETER? ? OR VARIABLE? ? OR METADATA OR META()DATA OR SIGNA-
             TURE? ? OR INTERFACE? ?)
S4    3318765   PROGRAM? ? OR OBJECT? ? OR CODE OR CLASS OR CLASSES OR MOD-
             ULE? ? OR APPLICATION? ? OR SOFTWARE OR FILE? ? OR METHOD? ? -
             OR ROUTINE? ? OR SUBROUTINE? ? OR THREAD? ?
S5     111593   S4(5N)(OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION? ? OR OPTI-
             MISATION? ? OR FINETUN??? OR FINE()TUN??? OR ENHANC??????)
S6      56045   (INCREAS??? OR AUGMENT? OR BOOST? OR AMPLIF? OR RAIS??? OR
             IMPROV??? OR RAMP??? OR UPGRAD?)(5N)S4(5N)(EFFICIENC??? OR EF-
             FICIENT OR EFFECTIV? OR SPEED)
S7      30794   FIRMWARE OR EMBEDDED()(CODE OR PROGRAM? ? OR SOFTWARE OR A-
             PPLICATION? ?)
S8         17   S1(100N)S5:S6(100N)S7
S9         44   S1(100N)S2:S3(100N)S7
S10        58   S8:S9
S11        22   S10 AND PY=1978:2003
S12        25   S10 AND AC=US/PR AND AY=(1978:2003)/PR
S13        25   S10 AND AC=US AND AY=1978:2003
S14        25   S10 AND AC=US AND AY=(1978:2003)/PR
S15        26   S11:S14
S16        26   IDPAT (sorted in duplicate/non-duplicate order)
```

16/3,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00695714
VIRTUAL GRAPHICS PROCESSOR AND METHOD FOR EMBEDDED, REAL TIME DISPLAY SYSTEMS
VIRTUELLER GRAPHIKPROZESSOR UND VERFAHREN FUR EINGEBETTETE ECHTZEITANZEIGESYSTEME
PROCESSEUR GRAPHIQUE VIRTUEL ET PROCEDURE POUR SYSTEMES D'AFFICHAGE EN TEMPS REEL INTEGRES
PATENT ASSIGNEE:
HONEYWELL INC., (246050), Honeywell Plaza, Minneapolis Minnesota 55408, (US), (applicant designated states: DE;FR;GB)
INVENTOR:
DENNISON, Tamara, G., 2255 Beacon Point Boulevard, Palm Harbor, FL 34683, (US)
FISHER, Jesse, G., 7815 Buckboard Northeast, Albuquerque, NM 87109, (US)
LINDSLEY, Steven, E., 8505 Gutierrez Northeast, Albuquerque, NM 87111, (US)
WEINGARTNER, Thomas, A., 1604 Propps Street Northeast, Albuquerque, NM 87112, (US)
LEGAL REPRESENTATIVE:
Rentzsch, Heinz, Dipl.-Ing. et al (9538), Honeywell Holding AG, Patent- und Lizenzabteilung, Kaiserleistrasse 39, 63067 Offenbach, (DE)
PATENT (CC, No, Kind, Date): EP 722589 A1 960724 (Basic)
EP 722589 B1 990120
WO 9510089 950413
APPLICATION (CC, No, Date): EP 94930439 940916; WO 94US10464 940916
PRIORITY (CC, No, Date): US 132727 931006
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS (V7): G06F-015/78;
NOTE:
No A-document published by EPO
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9903	818
CLAIMS B	(German)	9903	723
CLAIMS B	(French)	9903	984
SPEC B	(English)	9903	4017
Total word count - document A			0
Total word count - document B			6542
Total word count - documents A + B			6542

...SPECIFICATION video library) that contains graphics functions for a particular display system. By separating the hardware **dependency** from the means for generating the graphic pages, the designer need not know the intricacies...

...logic unit (also known as a screen engine) 30. The virtual graphics processor is an **embedded** **software** process, and these parts comprises data structures and internal processes within the overall process of...

...of the particular display system. These functions, along with static and dynamic data supplied as **arguments** thereto, are **passed** to the system's graphics hardware 31 for display of the graphic page and data...

16/3,K/16 (Item 16 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

01194983 **Image available**
METHOD FOR FIRMWARE VARIABLE STORAGE WITH EAGER COMPRESSION, FAIL-SAFE

EXTRACTION AND RESTART TIME COMPRESSION SCAN
METHODE POUR UN STOCKAGE VARIABLE DE MICROLOGICIEL PERMETTANT UNE COMPRESSION FACILE, UNE EXTRACTION A SECURITE INTEGREE ET UN BALAYAGE DE COMPRESSION DE DUREE DE REDEMARRAGE
Patent Applicant/Assignee:
INTEL CORPORATION, 2200 Mission College Boulevard, Santa Clara, CA 95052, US, US (Residence), US (Nationality)
Inventor(s):
ZIMMER Vincent, 1937 South 369th Street, Federal Way, WA 98003, US,
ROTHMAN Michael, 11905 183rd Street East, Puyallup, WA 98374, US,
MILLER Greg, 20505 NW Yoncalla Court, Portland, OR 972291, US,
DORAN Mark, 4605 33 rd Court NE, Olympia, WA 98516, US,
Legal Representative:
MALLIE Michael J (et al) (agent), Blakely Sokoloff Tayor & Zafman, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, CA 90025, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200502060 A2-A3 20050106 (WO 0502060)
Application: WO 2004US16585 20040526 (PCT/WO US04016585)
Priority Application: US 2003462996 20030616
Designated States:
(All protection types applied unless otherwise stated - for applications 2004+)
AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW
(EP) AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PL PT RO SE SI SK TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) BW GH GM KE LS MW MZ NA SD SL SZ TZ UG ZM ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Publication Language: English
Filing Language: English
Fulltext Word Count: 10087

Fulltext Availability:
Detailed Description

Detailed Description
... control is handed to DXE Dispatcher 102.
The DXE Dispatcher is responsible for loading and **invoking** DXE drivers found in **firmware** volumes, which correspond to the logical storage units from which firrnware is loaded under the ER framework. The DXE dispatcher searches for drivers in the **firmware** volumes described by the HOB List. As execution continues, other **firmware**
8
volumes might be located. When they are, the dispatcher searches them for drivers as...

...depends on the presence and contents of an a priori file and the evaluation of **dependency** expressions. These early DXE drivers will typically contain processor, chipset, and platform initialization code. These...

16/3,K/17 (Item 17 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT
(c) 2007 WIPO/Thomson. All rts. reserv.

01125551
METHOD, SYSTEM AND MEDIUM FOR CONTROLLING MANUFACTURE PROCESS HAVING MULTIVARIATE INPUT PARAMETERS
PROCEDE, SYSTEME ET SUPPORT PERMETTANT DE GERER UN PROCESSUS DE FABRICATION A L'AIDE DE PARAMETRES D'ENTREE MULTIDIMENSIONNELS
Patent Applicant/Assignee:

APPLIED MATERIALS INC, P.O. Box 450A, Santa Clara, CA 95052, US, US (Residence), US (Nationality), (For all designated states except: US)
Patent Applicant/Inventor:
KOKOTOV Yuri, Tzemech Hasadeh St. 50/7, 98533 Maaleh Adumim, IL, IL (Residence), IL (Nationality), (Designated only for: US)
ENTIN Efim, Aliyat Hanoar 26/7, 97234 Jerusalem, IL, IL (Residence), IL (Nationality), (Designated only for: US)
SEROR Jacques, Brandt St. 3/12, 93878 Jerusalem, IL, IL (Residence), IL (Nationality), (Designated only for: US)
FISHER Yossi, Mishol He'Hadas Street 3, 97277 Jerusalem, IL, IL (Residence), IL (Nationality), (Designated only for: US)
SAREL Shalomo, Maaleh Michmas 151, 90634 Maaleh Michmas, IL, IL (Residence), IL (Nationality), (Designated only for: US)
SHAMUGASUNDRAM Arulkumar P, 428 Madera Avenue #10, Sunnyvale, CA 94086, US, US (Residence), IN (Nationality), (Designated only for: US)
SCHWARM Alexander T, 9303 La Puente Drive, Austin, TX 78749, US, US (Residence), US (Nationality), (Designated only for: US)
PAIK Young Jeen, 4151 Acapulco Drive, Campbell, CA 95008, US, US (Residence), US (Nationality), (Designated only for: US)
Legal Representative:
KAHNG Anthony H (et al) (agent), Hale and Dorr LLP, The Willard Office Building, 1455 Pennsylvania Avenue, NW, Washington, DC 20004, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200446835 A2-A3 20040603 (WO 0446835)
Application: WO 2003US36501 20031114 (PCT/WO US03036501)
Priority Application: US 2002426393 20021115
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW
(EP) AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT RO SE SI SK TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) BW GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Publication Language: English
Filing Language: English
Fulltext Word Count: 5088

Fulltext Availability:
Detailed Description

Detailed Description
... the example implementation includes a number of components: an input transformer 401, an input-output **dependency** model 403, a corrector 405 and a storage device 407. All these components can be implemented in hardware, **firmware** , software and/or any combination thereof.

These components are further explained by also referring to...

...transformer 401 sends the transformed input values to the corrector 405.

The corrector 405, upon **receiving** the transformed input **parameter** values from the input transformer 401, sends the transformed input parameter values to the input/output dependence model 403. The input/output **dependency** model 403 then calculates predicted output parameter values Ypred (step 507). The corrector 405 then...

16/3,K/19 (Item 19 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

01004235 **Image available**
LIMITED TIME EVALUATION SYSTEM FOR FIRMWARE
SYSTEME D'EVALUATION EN TEMPS LIMITE POUR MICROLOGICIEL
Patent Applicant/Assignee:
PHOENIX TECHNOLOGIES LTD, 411 E. Plumeria Drive, San Jose, CA 95134, US, US (Residence), US (Nationality)
Inventor(s):
BRAMLEY Richard A Jr, 12 Gloria Drive, Mansfield, MA 02048, US,
Legal Representative:
NOBLES Kimberly G (agent), Irell & Manella LLP, Suite 400, 840 Newport Center Drive, Newport Beach, CA 92660, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200334210 A1 **20030424** (WO 0334210)
Application: WO 2002US30599 20020925 (PCT/WO US0230599)
Priority Application: US 2001976993 20011012
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LU MC NL PT SE SK TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Publication Language: English
Filing Language: English
Fulltext Word Count: 5439

Patent and Priority Information (Country, Number, Date):
Patent: ... **20030424**
Fulltext Availability:
Detailed Description
Publication Year: **2003**

Detailed Description
... principles of the'invention. For purposes of illustration, it is assumed that the time-dependent **firmware** feature of the present discussion is being installed into a personal computer (PC) system. In... for discussion purposes, the process 500 may be implemented and/or applied to any other **firmware** system.
The installation process 500 proceeds from a START state when the **routine** or **process** 500 is **called** . At **process** block 501, the BIOS **firmware** or code is copied into a temporary storage memory (such as memory 205 of Figure...

...determined, as shown in process block 503. In one embodiment, the end of the BIOS **firmware** or code is marked by some type of marker ...has been located, the time-dependent feature, along with corresponding code for checking the time- **dependency** , can be added to the end of the BIDS code (process block 505). In one embodiment, a second marker marking the end of the BIDS **firmware** (including the new feature and corresponding time- **dependency** checking code) can then be written/placed at the end of the code representing the global time-dependent feature and time
dependency checking code. The feature table within the BIOS can then be updated

16/3,K/22 (Item 22 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00805377 **Image available**

A METHOD AND SYSTEM FOR CIRCUMSCRIBING A TOPOLOGY TO FORM RING STRUCTURES
PROCEDE ET SYSTEME PERMETTANT DE CIRCONSCRIRE UNE TOPOLOGIE POUR FORMER DES STRUCTURES ANNULAIRES

Patent Applicant/Assignee:
SONY ELECTRONICS INC, 1 Sony Drive, Park Ridge, NJ 07656, US, US (Residence), US (Nationality)
Inventor(s):
JAMES David V, 3180 South Court, Palo Alto, CA 94306, US,
SHIMA Hisato, 707 Continental Circle, No. 237, Mountain View, CA 94040, US,
STONE Glen David, 1501 Oburn Court, Campbell, CA 95008, US,
FAIRMAN Bruce, 275 Martinez Road, Woodside, CA 94062, US,
Legal Representative:
SOBRINO Maria McCormack (et al) (agent), Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025-1026, US,

Patent and Priority Information (Country, Number, Date):
Patent: WO 200138951 A2-A3 **20010531** (WO 0138951)
Application: WO 2000US42412 20001129 (PCT/WO US0042412)
Priority Application: US 99167958 19991129; US 2000723920 20001127
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Publication Language: English
Filing Language: English
Fulltext Word Count: 12842

Patent and Priority Information (Country, Number, Date):
Patent: ... **20010531**
Fulltext Availability:
Detailed Description
Publication Year: **2001**

Detailed Description

... another. These messages are designed to be independent, in that a sequence number allows the **receiver** to determine when a duplicate **message** was sent (typically due to a fault) and should be discarded. All of these message...

...values can be safely reused before the normal SPLIT-TIMEOUT timeout delay.

To avoid circular- **dependency** deadlocks, one portal in this circular lists of portals **receives** the **message** in a logical "request" queue and outputs a message (to the next portal) on a second logical "response" queue. Mes(section)age processing hardware/ **firmware** allows response queue messages to be processed ahead of previously accepted request

```
File 275:Gale Group Computer DB(TM) 1983-2007/Jun 08
         (c) 2007 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2007/Jun 08
         (c) 2007 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2007/Jun 01
         (c) 2007 The Gale Group
File  16:Gale Group PROMT(R) 1990-2007/Jun 08
         (c) 2007 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2007/Jun 08
         (c)2007 The Gale Group
File 624:McGraw-Hill Publications 1985-2007/Jun 06
         (c) 2007 McGraw-Hill Co. Inc
File  15:ABI/Inform(R) 1971-2007/Jun 12
         (c) 2007 ProQuest Info&Learning
File 647:CMP  Computer Fulltext 1988-2007/Sep W1
         (c) 2007 CMP Media, LLC
File 674:Computer News Fulltext 1989-2006/Sep W1
         (c) 2006 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2007/Jun 11
         (c) 2007 Dialog
File 369:New Scientist 1994-2007/Jan W1
         (c) 2007 Reed Business Information Ltd.

Set     Items   Description
S1     442142   DAG OR DIGRAPH OR GRAPH? ? OR DEPENDENCY OR CALL()(DIAGRAM?
              ? OR GRAPH? ? OR CHART? ?) OR TOPOLOGICAL()ORDER???
S2     992917   (CALL??? OR INVOK??? OR INVOCATION)(7N)(PROGRAM? ? OR OBJE-
             CT? ? OR CODE OR CLASS OR CLASSES OR MODULE? ? OR APPLICATION?
              ? OR SOFTWARE OR FIRMWARE OR FILE? ? OR METHOD? ? OR ROUTINE?
              ? OR SUBROUTINE? ? OR THREAD? ? OR PROCESS OR PROCESSES)
S3     138704   (PASS OR PASSES OR PASSING OR PASSAGE OR PASSED OR STAMP???
             OR COUPL??? OR RECEIV???)(5N)(MESSAGE? ? OR ARGUMENT? ? OR P-
             ARAMETER? ? OR VARIABLE? ? OR METADATA OR META()DATA OR SIGNA-
             TURE? ? OR INTERFACE? ?)
S4   18635220   PROGRAM? ? OR OBJECT? ? OR CODE OR CLASS OR CLASSES OR MOD-
             ULE? ? OR APPLICATION? ? OR SOFTWARE OR FIRMWARE OR FILE? ? OR
              METHOD? ? OR ROUTINE? ? OR SUBROUTINE? ? OR THREAD? ?
S5     609678   S4(5N)(OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION? ? OR OPTI-
             MISATION? ? OR FINETUN??? OR FINE()TUN??? OR ENHANC??????)
S6     167742   (INCREAS??? OR AUGMENT? OR BOOST? OR AMPLIF? OR RAIS??? OR
             IMPROV??? OR RAMP???, OR UPGRAD?)(5N)S4(5N)(EFFICIENC??? OR EF-
             FICIENT OR EFFECTIV? OR SPEED)
S7      99250   FIRMWARE OR EMBEDDED()(CODE OR PROGRAM? ? OR SOFTWARE OR A-
             PPLICATION? ?)
S8         85   S1(100N)S5:S6(100N)S7
S9         47   RD  (unique items)
S10        35   S9 NOT PY=2004:2007
```

10/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01463984 SUPPLIER NUMBER: 11609526 (USE FORMAT 7 OR 9 FOR FULL TEXT)
C++ solution for embedded SPARC systems. (Microtec Research Inc.'s SPARCompiler C++ 3.0)(Brief Article) (Product Announcement)
C Users Journal, v9, n12, p128(1)
Dec, 1991
DOCUMENT TYPE: Product Announcement ISSN: 0898-9788 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 201 LINE COUNT: 00017

... compiler implements the 2.1 definition of the C++ language with numerous extensions to support **embedded applications** development. Complementing the compiler is the Microtec XRAY Debugger, **enhanced** for debugging **optimized** C++ **code** . Microtec Research will also supply a variety of inspection and conversion tools, including the OWL...

...Research Capsule Class Library, a collection of class library building blocks for data structure management, **graph** manipulation, and error handling.
Microtec Research will initially offer the SPARC C++ Compiler and companion...

10/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01356019 SUPPLIER NUMBER: 08328564 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Optimizing compilers struggle to meet the challenge of silicon.
Williams, Tom
Computer Design, v29, n7, p67(6)
April 1, 1990
ISSN: 0010-4566 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 4122 LINE COUNT: 00323

... version for the Motorola 88000, according to David Glass, QTC senior product marketing engineer. The **optimizer** reads in the assembly **code** generated by a third-party compiler and schedules it to take advantage of parallel execution...

...hand-tune what the automatic mode has delivered. One part of the screen displays the **code** as the scheduler has **optimized** it, and the other window displays data dependencies. Users can query operations and move them around to improve scheduling. The optimizer will alert users in case conflicts or data **dependency** problems arise.
The QTC optimizer doesn't produce any specific debugging information, so it can...
...minds are working on these problems, perfection is still far off when it comes to **optimizing code** on RISC machines, especially for **embedded applications** . The silicon technology has moved faster than the software technology.
Toolmakers for advanced processors are...

10/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01317892 SUPPLIER NUMBER: 07893504 (USE FORMAT 7 OR 9 FOR FULL TEXT)
ADS v. 5.1; Aion Corp. (Software Review) (Software Review) (evaluation)
Glenwright, Joy C.
AI Expert, v4, n11, p65(7)

Nov, 1989
DOCUMENT TYPE: evaluation ISSN: 0888-3785 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3923 LINE COUNT: 00308

... applications, and object-oriented programming has matured. Reflecting these trends, Aion ADS 5.1 includes **object** orientation as well as **enhanced** inference-based functions.
Using expert-systems technology in traditional and inference-based applications has become...

...Teradata, DB2) and indirectly (IDMS, FOCUS). In addition, we've developed both stand-alone and **embedded applications** .
The new version of ADS is a boon to developers. It contains both inference-based...

...major features I'll examine are the new on-line screen facilities, inference engine changes, **object** orientation, and **enhanced** explanation capabilities.
Like the previous version, ADS 5.1's knowledge base is developed by creating knowledge-base objects. The original objects: state, parameter, type, rule, function, process, message, report, **graph** , and vocabulary are joined by new objects: display, group, class, slot, method, and instance. These...

10/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2007 The Gale Group. All rts. reserv.

01119612 SUPPLIER NUMBER: 00623810
The Tandy 200.
Hartmann, T.
Popular Computing, v4, n9, p82-86
July, 1985
DOCUMENT TYPE: evaluaton ISSN: 0279-4721 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

...ABSTRACT: ROM bank is limited to 19K bytes, it accepts up to 32K bytes of additional **firmware** . The system's built-in software includes Multiplan, Microsoft BASIC, Telcom communications **program** , an **enhanced** text editor, address organizer, scheduler, alarm, and calculator. This is a solid, elegant machine that...

...competitors within its price category of $999. A photograph illustrates the Tandy 200, and a **graph** shows its features at a glance.

10/3,K/5 (Item 1 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2007 The Gale Group. All rts. reserv.

04020550 Supplier Number: 131629595 (USE FORMAT 007 FOR FULLTEXT)
InterDesign Technologies Selects ILOG Views to Enhance Product Line User Interface.
PR Newswire, pNA
Dec 9, 2003
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 508

... ILOG Views to help customers more effectively leverage virtual prototype environments in order to realize **firmware** validation before creating test models.

-- Venet: Vehicle Network System Simulator -- Venet, a vehicle network system simulator, utilizes ILOG Views including **Graph** Layout to enable

advanced network system modeling and simulation, which is achieved vi a

CAN (Controller...
...of their products.

About ILOG

For more than 10 years, ILOG's innovative enterprise-class **software** components and services have helped companies maximize their business agility and **improve** operating **efficiency** . Over 1000 global corporations and more than 300 leading **software** vendors rely on ILOG's business rules, optimization and visualization technologies to achieve dramatic returns...

10/3,K/6 (Item 2 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

03585412 Supplier Number: 110742522 (USE FORMAT 007 FOR FULLTEXT)
DoubleWide Software Unleashes Power of Embedded Supersystems with Introduction of DoubleWide Studio.
Business Wire, p5546
Dec 2, 2003
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 822

... Software(TM), Inc., the pioneer in Embedded Supersystems Virtualization, today announced a significant improvement in **embedded software** development with the introduction of DoubleWide Studio(TM), groundbreaking software that dramatically reduces the build...

...broad range of industry segments including networking equipment, aerospace, industrial and medical devices to dramatically **increase** the **speed** and **efficiency** of **software** development and testing by virtualizing multiple, complex embedded systems.

DoubleWide Studio offers unique "Embedded Supersystems...

...market while decreasing prototype costs."

DoubleWide customers, including Foundry Networks (Nasdaq:FDRY), benefit from reduced **dependency** on hardware prototypes and realize dramatic reductions in associated prototype costs. DoubleWide Studio is an ...

10/3,K/7 (Item 3 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

03541518 Supplier Number: 108552752 (USE FORMAT 007 FOR FULLTEXT)
AppIQ Introduces the Most Comprehensive Solution for Integrated SAN Management, Storage Resource Management, and Storage Operations Management.
Business Wire, p5422
Oct 6, 2003
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 1902

... and directors
-- EMC CLARiiON disk arrays with PowerPath
-- Raw volumes

-- Additional Brocade and LSI Logic **firmware** revisions
StorageAuthority is also the first and only solution that manages SGI's recently introduced...

...data and improve operational efficiency.
Rules-Based, End-to-End Provisioning
The new StorageAuthority Provisioning **module** **enhances** AppIQ's end-to-end provisioning capabilities with a rules engine that simplifies, secures, and...

...standard. Offers
auto-discovery of hosts, HBAs, SAN switches, and disk subsystems; graphical topology mapping; **dependency** and path management; capacity and performance reports; event management; trending; policy-based automation; and role-based security. Built-in Advisors and Automators simplify complex tasks such as replacing HBAs; upgrading **firmware** ; understanding what users and data are impacted by planned or unplanned downtime; and identifying new...

10/3,K/8 (Item 4 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

03140472 Supplier Number: 83728475 (USE FORMAT 007 FOR FULLTEXT)
Green Hills Software Launches Version 4.0 of INTEGRITY Real-Time Operating System; Now Most Powerful, Testable and Reliable RTOS in the Industry.
Business Wire, p0040
March 13, 2002
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 1173

... O device and networking support.
INTEGRITY is a secure, fast, deterministic, real-time operating system **optimized** for **embedded** **applications** that place a premium on reliability, real-time performance, and testability. Utilizing a processor's...

...based analyzer displays this information for a given point in time, or as a line **graph** , providing both a snapshot and historical view of CPU time and memory use.
INTEGRITY's...

10/3,K/9 (Item 5 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

03137959 Supplier Number: 83655274 (USE FORMAT 007 FOR FULLTEXT)
Hitachi's Integrated Development Environment Makes it Easier for Customers to Write Software for Embedded Control Applications.
Business Wire, p2090
March 11, 2002
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 901

... and MPUs for greater efficiency.
Besides providing the compiler, assembler and linker tools needed for **optimized** **software** builds, and the simulator/debugger for testing the application code produced in the build process, the HEW2 environment has functions that engineers can use to create and manage entire **embedded** **software** projects. It offers tools for project creation, file management,

version control, source code editing, control of software build configurations, **dependency** checking, and toolchain configuration control.
Wizards simplify start-up processes
System engineers can start projects...

...guide them through the selection of configuration options, debugger targets and the creation of startup **code** . An intermodule **optimizing** linker produces better **optimized** **software** load **modules** . Also, the editor is **enhanced** , primarily to support debugging; break functions are improved; and the toolchain options are integrated into...

10/3,K/10 (Item 6 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2007 The Gale Group. All rts. reserv.

01596151 Supplier Number: 48212072 (USE FORMAT 007 FOR FULLTEXT)
Introducing the CodeTRACE(TM) Source-Level Trace Analysis Tool for Embedded Software Developers
PR Newswire, p0106SFTU077
Jan 6, 1998
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 675

... display the values contained in source-level variables or combinations.
CodeTRACE provides the ability to **graph** the execution of specific functions and data structures, allowing users to rapidly locate regions of ...

...and is priced at U.S. $2,000.
About Applied Microsystems
The leader in hardware- **enhanced** **embedded** **software** design, debug and test solutions, Applied Microsystems helps engineers develop products faster, more reliably and...
...engineers, to powerful full-scale models for both software and hardware development; and category-defining **embedded** **software** verification tools.
The company can be reached at P.O. Box 97002, Redmond, Wash. 98073...

10/3,K/11 (Item 7 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2007 The Gale Group. All rts. reserv.

01541814 Supplier Number: 47449412 (USE FORMAT 007 FOR FULLTEXT)
RadiSys and Intrinsyc announce embedded software alliance.
Business Wire, p06090338
June 9, 1997
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 860

... meet our customers' needs for complete Win32-based solutions."
Alex Doumani, general manager of RadiSys' **Embedded** **Software** Operation, stated, "Intrinsyc is developing technologies and products that complement RadiSys' strategy for embedded Windows...

...creation and deployment of minimal footprint applications. The key features of IX include automatic application **dependency** analysis, source code profiling, OS configuration and target generation. The analysis is dynamic, requires no...

...and works for any version of Windows NT or CE. Developers can use IX to

improve the efficiency of their **code** , **optimize** compiler flags, minimize memory requirements and manage a set of programs/libraries that comprise a...

10/3,K/12 (Item 8 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01228371 Supplier Number: 44022170 (USE FORMAT 007 FOR FULLTEXT)
NEW DESIGNMATE PLOTTER FROM CALCOMP OUTPUTS A- TO E-SIZE FULL-COLOR DRAWINGS AND COSTS LESS THAN $3,000
News Release, p1
August 9, 1993
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 752

... drawings, tool designs, assembly
drawings. schematics, printed circuit board designs project management timelines, process control **graphs** and business presentations.
Performance and Productivity
The DesignMate 3036 can plot as fast as 42...

...per second (ips)
at an acceleration of up to 2.8 g. Total throughput is **enhanced** by CaIComp's Plot Manager **firmware**
, which minimizes pen movement and pen changes.
Resolution is 0.0005 inches (or 2,032...

10/3,K/13 (Item 9 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01192828 Supplier Number: 42988590 (USE FORMAT 007 FOR FULLTEXT)
GENESIS QA2500 Sets New Standards For Data Collector Performance and Capability
News Release, p1
May 12, 1992
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 264

... And a new, "tactile-feel," Qwerty keyboard makes user input and programming easy, fast, and **effective** .

With the **improved** intuitive **firmware** , the user can now **speed** through attribute and/or variable data collecting and reporting with minimal keystrokes. And an impressive package of charts, **graphs** , and data displays actively communicate process status and identify corrective action paths.

The GENESIS QA2500...

10/3,K/14 (Item 10 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01172397 Supplier Number: 42319970 (USE FORMAT 007 FOR FULLTEXT)
Microtec Research Announces First C++ Solution for Embedded SPARC Systems

News Release, p1
August 30, 1991
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 727

... Compiler implements the 2.1 definition of the C++ language with numerous extensions to support **embedded applications** development. Complementing the compiler is the popular Microtec XRAY Debugger (TM), **enhanced** for debugging **optimized** C++ **code** . Microtec Research will also supply a variety of inspection and conversion tools, including the OWL...

...Research Capsule Class Library, a collection of class library building blocks for data structure management, **graph** manipulation, and error handling.

The XRAY C++ source-level debugger for SPARC features a flexible... ...a unique, tightly-coupled tool chain, allowing embedded systems engineers to develop and debug highly **optimized** C++ SPARC **applications** . "The debugging of **optimised** production **code** is key to embedded users of the SPARC architecture" said Jim Ting, Business Development SPARC...

10/3,K/15 (Item 11 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2007 The Gale Group. All rts. reserv.

01168096 Supplier Number: 42217609 (USE FORMAT 007 FOR FULLTEXT)
FULL COLOR, INTERACTIVE TOUCH SCREEN INDUSTRIAL SEALED GRAPHICS MODULE
News Release, p1
July 15, 1991
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 342

... display. Utilizing recent advances in Flash EPROM technology, the M6ST is capable of receiving future **software enhancements** and additions to the **firmware** in the field.

The product offers the user common text emulation and industry standard ReGIS...

...ASCll based protocol with extensive commands for graphics including circles, arcs, pies, vectors, multiple vector **graphs** , concave or convex area fills, paint, etc.

An independent micro processor manages the (80x60) interactive IR touch array providing immediate responses to operator interaction. The touch **firmware** in the M6ST provides 128K bytes of button definition area. The touch processor rapidly scans...

10/3,K/16 (Item 12 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2007 The Gale Group. All rts. reserv.

01143189 Supplier Number: 41325215 (USE FORMAT 007 FOR FULLTEXT)
MICROTEC RESEARCH INC. INTRODUCES PERFORMANCE ANALYSIS AND CODE COVERAGE TOOL
News Release, p1

May 9, 1990
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 430

... member of the XRAY Debugger (TM)
family which provides performance and code coverage analysis of
embedded applications
. Named XRAY/DX the product is available for the
Motorola 680x0 microprocessor family.

XRAY/DX...

...level source code. As with all Microtec Research
XRAY products, XRAY/DX supports analysis of **optimized** , production
quality **code** .

XRAY/DX extends and **enhances** XRAY's set of debugging commands. The
product features an easy-to-learn window-oriented...

...and
debugged at the same time. XRAY/DX can generate ten analysis reports
which include: **call graphs** , function/subroutine execution timings,
frequency of function invocation, duration of function invocation,
statements/instructions executed...

10/3,K/17 (Item 13 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2007 The Gale Group. All rts. reserv.

01126459 Supplier Number: 41008698 (USE FORMAT 007 FOR FULLTEXT)
READY SYSTEMS INTRODUCES CARDtools 3.6
News Release, p1
Nov 1, 1989
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 783

... customers a better solution for Ada and government
applications, as well as general real-time **embedded applications**
. Our
customers can now have a complete real-time CASE solution that
interfaces with their...

...task priorities,
and synchronization are managed by TaskTimer and graphically
represented in Task Maps.

TaskTimer **graphs** the timing analysis of individual paths and the
interaction of those paths within the software...
...allows the software developer to link and trace
software requirements forwards and backwards in the **software**
development lifecycle. **Enhanced** capabilities provide flexibility and
openness, allowing TraceBuilder II to track requirements through
documents and files...

10/3,K/18 (Item 14 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2007 The Gale Group. All rts. reserv.

01059440 Supplier Number: 40234259 (USE FORMAT 007 FOR FULLTEXT)
INTEL OFFERS ADA (TM) SUPPORT FOR 80386
News Release, p1

Dec 11, 1987
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 543

... s Development Tools
Operation. "Intel's Ada-386 provides a full set of language tools **optimized** for the 80386 architecture. Ada **applications** developers can
now come to one vendor, Intel, for the whole Ada development solution -- everything...

...the code
is executing on the target.

A Global Optimizer that reduces the size and **increases** execution **speed** of **embedded application code** .

A Linker that combines separately-compiled modules and Ada runtime routines into an executable image...

...different
hardware environments.

Other language tools to enhance programmer productivity, including a Cross Referencer, Source **Dependency** Lister and Source Formatter.

Features and Capabilities

The new Ada development tools are designed to help the Ada **embedded applications** programmer create efficient, high-performance 80386 code. Key features include:

Built-in support for the...

...INTERFACE to Intel's ASM-386, PL/M-386 and iC-386.

In addition, the **embedded applications** programmer will benefit from
built-in interrupt handling optimizations and the modular nature of the...

10/3,K/19 (Item 1 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

05687599 Supplier Number: 111068310 (USE FORMAT 7 FOR FULLTEXT)
Interdesign Technologies selects ILOG views to enhance user interface for product line.
M2 Presswire, pNA
Dec 9, 2003
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 687

... ILOG Views to help customers more effectively leverage virtual prototype environments in order to realise **firmware** validation before creating test models.
* Venet: Vehicle Network System Simulator - Venet, a vehicle network system simulator, utilises ILOG Views including **Graph** Layout to enable advanced network system modeling and simulation, which is achieved via CAN (Controller...

...of their products.

ABOUT ILOG
For more than 10 years, ILOG's innovative enterprise-class **software** components and services have helped companies maximise their business agility and **improve** operating **efficiency** . Over 1000 global corporations and more than 300 leading **software** vendors rely on ILOG's business rules, optimisation and visualisation technologies to achieve dramatic returns...

10/3,K/20 (Item 2 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

03204414 Supplier Number: 46568252 (USE FORMAT 7 FOR FULLTEXT)
MOTOROLA: Motorola provides system solutions for the digital set top box market
M2 Presswire, pN/A
July 24, 1996
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 1433

... complete solution to the front end of the digital set-top box reduces the manufacturers' **dependency** on multiple suppliers as well as offering less PCB (printed circuit board) real estate. It...

...tremendous amount of investment has already been made both from the broadcasters and manufacturers to **optimize** **software** running on these microprocessors, Motorola will introduce throughout this year a series of higher performance...

...RISC: microprocessor performance without the code expansion normally experienced by using RISC based technology in **embedded** **applications** like DSTB. This allows good cost trade off between code density, performance and memory savings...

10/3,K/21 (Item 3 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

01003884 Supplier Number: 40281086 (USE FORMAT 7 FOR FULLTEXT)
INTEL MAKES 80386 ATTRACTIVE TO MILITARY APPLICATIONS
Automated Manufacturing Strategy, v9, n4, pN/A
Feb, 1988
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 466

... while the code is executing on the target. A global optimizer reduces the size and **increases** the execution **speed** of the **embedded** **application** **code** .
A linker combines separately-compiled modules and Ada runtime routines into an executable image. An...

...different hardware environments. Other language tools which enhance programmer productivity include a cross referencer, source **dependency** lister and source formatter.
The new Ada development tools are designed to help the Ada **embedded** **applications** programmer create an efficient, high-performance 80386 code. Key features include built-in support for...

10/3,K/22 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

09652961 Supplier Number: 84072885 (USE FORMAT 7 FOR FULLTEXT)
GREEN HILLS SOFTWARE LAUNCHES 4.0 OF INTEGRITY REAL-TIME OS.(Product Announcement)
Software Industry Report, v34, n6, p5
March 18, 2002
Language: English Record Type: Fulltext
Article Type: Product Announcement
Document Type: Newsletter; Trade
Word Count: 993

... O device and networking support.
INTEGRITY is a secure, fast, deterministic, real-time operating system **optimized** for **embedded** **applications** that place a premium on reliability, real-time performance, and testability. Utilizing a processor's...

...based analyzer displays this information for a given point in time, or as a line **graph** , providing both a snapshot and historical view of CPU time and memory use.
INTEGRITY's...

10/3,K/23 (Item 2 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

07054760 Supplier Number: 58467961 (USE FORMAT 7 FOR FULLTEXT)
Agfa to Launch New Sherpa Line: In Bid for Digital Proofing Leadership.(Product Development)
Tribute, Andrew
The Seybold Report on Publishing Systems, v29, n1, pNA
Sept 1, 1999
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 2137

... debut on Sept. 20 at IGAS in Tokyo. A U.S. introduction will follow at **Graph** Expo in October.
Sherpa. The current Sherpa product was introduced in January and has been...

...s specifications. Areas of change include improvements to the mechanics to achieve greater imaging accuracy, **enhancements** to the engine **firmware** and the provision of finer color control through the addition of Agfa's color management...

...thus reducing the benefit of the finer positioning. To rectify that situation, the new engine **firmware** will reduce the amount of ink that is released with each spot, thus reducing the...

10/3,K/24 (Item 3 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

05152002 Supplier Number: 47863736 (USE FORMAT 7 FOR FULLTEXT)
Real-Time Software -- Software supports Win32
Costlow, Terry
Electronic Engineering Times, p76
July 28, 1997
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 317

... using Windows CE in embedded products.
Integration Expert takes on several jobs for developers of **embedded software** . First, it does an automatic **dependency** analysis, which looks at the static and dynamic relationships between the application package and the operating system. Next, it does profiling to calculate the target footprint and **improve code efficiency** and configuration, a task that provides an information model to represent the OS and hardware...

...are done, Integration Expert will generate target code.
A mainstay of the line is its **dependency** analysis. This feature has a number of benefits even for designers who aren't developing real-time code.
"The automatic **dependency** analysis occurs regardless of which operating system you use," said Brian Rose, marketing director at...

10/3,K/25 (Item 4 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

01865018 Supplier Number: 42365023 (USE FORMAT 7 FOR FULLTEXT)
Embedded C+ + tools emerge
Electronic Engineering Times, p64
Sept 16, 1991
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 283

... from Sun Microsystems Inc.
The ANSI-compatible C+ + cross-compiler includes numerous extensions to support **embedded applications** development, including re-entrant, ROMable code generation; support for the ANSI "volatile" key word; multiple ...

...the Capsule Class Library, a collection of class-library building blocks for data-structure management, **graph** manipulation, and error handling.
The XRAY C+ + source-level debugger for Sparc provides a window...

...declared, as well as C+ + type and inheritance information. The XRAY breakpoint facility has been **enhanced** to support complex breakpoints on **classes** , object instances, and overloaded functions. The debugger works with instruction-set simulation, in-circuit emulation, and in-circuit **firmware** monitors.
The Sparc C+ + tools are initially available on Sparc platforms, with other hosts--including...

10/3,K/26 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

12559281 SUPPLIER NUMBER: 65091397 (USE FORMAT 7 OR 9 FOR FULL TEXT)
IDE tools from (micro)C vendors speed embedded-application development.(Product Announcement)
Nath, NS Manju
EDN, 45, 17, 22
August 17, 2000
DOCUMENT TYPE: Product Announcement ISSN: 0012-7515 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 905 LINE COUNT: 00075

... architecture is unique, finding acceptance for a device in the market is difficult. As an **embedded - application** developer, you are duty-bound to select the device that has the best tool-design...

...ICE 2000 Emulator and PICMaster in-circuit emulator.
Using MPLab, you can write, debug, and **optimize** the PICmicro (micro)C **applications** . The MPLab Project Manager organizes the files that include your application **firmware** under one project. It also allows you to create a project; add, edit, or debug...

...source, dependencies, and object fries as a graphic tree. You can also automatically update the **dependency** list in the tree. The language-sensitive editor uses colors to display the instruction, assembler ...

10/3,K/27 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2007 The Gale Group. All rts. reserv.

11419251 SUPPLIER NUMBER: 56176666 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Perk up productivity with profiling tools.(evaluation of debugging/testing software; includes related article on hardware-assisted software profiling)(Statistical Data Included)
Nath, NS Manju
EDN, 44, 18, 123
Sept 2, 1999
DOCUMENT TYPE: Statistical Data Included ISSN: 0012-7515
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 3204 LINE COUNT: 00271

... code execution. For example, during the trace-execution function, the tool performs a trace on **embedded software** , not hardware. A software "instrumenter" utility prepares the user's program for in-circuit verification...

...task, function, or branch level. Using the tool's list of performance characteristics, you can **optimize** select portions of the **code** .
CodeTEST also includes the CodeTEST-Memory, which tracks dynamic-memory allocation in **embedded programs** . Because the tool time-stamps each line of the trace, you can identify how long...
...portions of code that would execute during a given test scenario. An interactive X-Y **graph** shows the extent of coverage the tool achieves over the testing time.
Another element of...

10/3,K/28 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2007 The Gale Group. All rts. reserv.

09728563 SUPPLIER NUMBER: 19756110 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Annasoft Inks Distribution Pact for Intrinsyc's Embedded CE Tools and Components.
Business Wire, p9160086
Sep 16, 1997
LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 948 LINE COUNT: 00090

... Annasoft distribute our full product line. As the preeminent supplier of Microsoft operating systems for **embedded applications** and the founding member of the Windows CE Embedded Initiative, Annasoft will provide our products...

...creation and deployment of minimal footprint applications. The key features of IX include automatic application **dependency** analysis, source code profiling, OS configuration, and target generation. The analysis is dynamic, requires no...

...Win32-based operating systems, including Windows 95, NT and CE. Developers can use IX to **improve** the efficiency of their **code** , **optimize** compiler flags, minimize memory requirements, and manage a set of programs/libraries that comprise a...

10/3,K/29 (Item 4 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

05810267 SUPPLIER NUMBER: 11865454 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Firmware improves 10-to-2000 MHz noise-figure meter: a venerable test tool is resurrected with a new set of measurement firmware for enhanced accuracy and ease of use.
Microwaves & RF, v31, n1, p142(1)
Jan, 1992
ISSN: 0745-2993 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 482 LINE COUNT: 00038

... a new manufacturer: Maury Microwave Corp. (Ontario, CA). It also has a new set of **firmware** that **enhances** performance and accuracy while making it easy to use. The MT2075B measures the noise figure...

...Microwaves & RF, July 1991, p. 23), Maury's engineers engaged in an intensive firmwave development **program** to **enhance** the performance of what is now known as the MT2075B.
These firmwave improvements include the...

...Earlier, clearing such messages required resetting the instrument. Also, heads and scales for digital-plotter **graphs** can be customized. The new software simplifies entry into, editing, and exit from the insertion...
...their instruments refurbished to the performance levels of an MT2075B by means of an MT2075B30 **firmware** upgrade pack (Fig. 2). The pack consists of two erasable programmable memory (EPROM) chips with...

10/3,K/30 (Item 5 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

03711370 SUPPLIER NUMBER: 06860834 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Picture clears for industrial touch screens.
Leland, Brad
Machine Design, v60, n28, p93(4)
Nov 24, 1988
ISSN: 0024-9114 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 2098 LINE COUNT: 00172

... use.
Software development aids may also simplify the programming needed to generate the charts and **graphs** that often accompany touch screens controlling real-time processes . One development aid, for example, simplifies...

...each time the host sends the percentage value of full scale for the bar chart.
Software -configurable character sets are another **enhancement** to look for in development aids. This feature is useful in quickly generating a custom...

...in blanks. ScreenBuilder generates all ANSI screen -construction commands and can download them into the **firmware** of Fluke 1030/50 subsystems when complete. A related package, called IconBuilder, allows development of...

10/3,K/31 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

01357761 00-08748
RAID performance parameters and optimizing features
Regester, Mark
Computer Technology Review v16n11 PP: 38, 42+ Nov 1996
ISSN: 0278-9647 JRNL CODE: CTN
WORD COUNT: 1877

...TEXT: by increasing the amount of memory available for storing pre-fetch requests.

Write-Back Caching

(**Graph** Omitted)

Captioned as: Fig 1

(**Graph** Omitted)

Captioned as: Fig 2

Just as a read cache can significantly improve read response...
...not only provides better performance through lower response times, it also gives the array controller **firmware** the opportunity to **optimize** the movement of data from cache to disk media. The most important optimization is the...

10/3,K/32 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00288159 85-28593
Business Users Ponder Desktop Pen Plotters/Desktop Plotters
Dalrymple, Rick
Mini-Micro Systems v18n10 PP: 157-174 Jul 1985
ISSN: 0364-9342 JRNL CODE: MOD

ABSTRACT: Inexpensive desktop plotters featuring **firmware** that responds to high-level control sequences were introduced by pen plotter manufacturers for technical users. However, the desktop plotter has not been as popular for business **applications** . To **enhance** the appearance of **graphs** drawn either by spreadsheet software or pen plotters, business users are turning to **software programs** that interactively **enhance** the **graph** drawn by these **methods** , such as the Grafix Partner from Brightbill-Roberts and Co. Ltd. Computer-image recorders allow...

10/3,K/33 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

01133323 CMP ACCESSION NUMBER: EET19970728S0080
Real-Time Software - Software supports Win32
Terry Costlow
ELECTRONIC ENGINEERING TIMES, 1997, n 964, PG76
PUBLICATION DATE: 970728
JOURNAL CODE: EET LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: Design
WORD COUNT: 317

... using Windows CE in embedded products.
Integration Expert takes on several jobs for developers of **embedded software** . First, it does an automatic **dependency** analysis, which looks at the static and dynamic relationships between the application package and the operating system. Next, it does profiling to calculate the target footprint and **improve code efficiency** and configuration, a task that provides an information model to represent the OS and hardware...

...are done, Integration Expert will generate target code.
A mainstay of the line is its **dependency** analysis. This feature has a number of benefits even for designers who aren't developing real- time code.
"The automatic **dependency** analysis occurs regardless of which operating system you use," said Brian Rose, marketing director at...

10/3,K/34 (Item 2 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

00607317 CMP ACCESSION NUMBER: UNX19911014S2728
C++ For SPARC Made Available
DAVID FIEDLER
UNIX TODAY , 1991, n 082, 24
PUBLICATION DATE: 911014
JOURNAL CODE: UNX LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: products
WORD COUNT: 283

... I/O streams, as well as class library functions for error handling, data structures and **graph** manipulation.
The XRAY debugger is a sophisticated software tool that allows engineers to debug C...

...is window- oriented and available in several versions for use with in-circuit emulation and **firmware** monitors.
XRAY has knowledge of subtle C++ addressing and data type nuances and is tightly coupled to the C++ compiler, so that debugging of **optimized code** is possible. Conversely, setting the debug switch on the compiler doesn't make the resulting...

10/3,K/35 (Item 3 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

00593225 CMP ACCESSION NUMBER: EET19910916S2436
Embedded C++ tools emerge
RICHARD GOERING
ELECTRONIC ENGINEERING TIMES, 1991, n 659, 64
PUBLICATION DATE: 910916
JOURNAL CODE: EET LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: Design: CAE Tools
WORD COUNT: 291

... the Capsule Class Library, a collection of class-library building blocks for data-structure management, **graph** manipulation, and error handling.
The XRAY C++ source-level debugger for Sparc provides a window...

...declared, as well as C++ type and inheritance information. The XRAY

breakpoint facility has been **enhanced** to support complex breakpoints on **classes** , object instances, and overloaded functions. The debugger works with instruction-set simulation, in- circuit emulation, and in-circuit **firmware** monitors.

```
File   8:Ei Compendex(R) 1884-2007/Jun W1
         (c) 2007 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2007/May
         (c) 2007 ProQuest Info&Learning
File  65:Inside Conferences 1993-2007/Jun 12
         (c) 2007 BLDSC all rts. reserv.
File   2:INSPEC 1898-2007/Jun W1
         (c) 2007 Institution of Electrical Engineers
File   6:NTIS 1964-2007/Jun W3
         (c) 2007 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2007/Jun W1
         (c) 2007 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 2006 The Thomson Corp
File  34:SciSearch(R) Cited Ref Sci 1990-2007/Jun W2
         (c) 2007 The Thomson Corp
File  99:Wilson Appl. Sci & Tech Abs 1983-2007/May
         (c) 2007 The HW Wilson Co.
File 266:FEDRIP 2007/May
         Comp & dist by NTIS, Intl Copyright All Rights Res
File  95:TEME-Technology & Management 1989-2007/Jun W2
         (c) 2007 FIZ TECHNIK
File  56:Computer and Information Systems Abstracts 1966-2007/May
         (c) 2007 CSA.
File  60:ANTE: Abstracts in New Tech & Engineer 1966-2007/May
         (c) 2007 CSA.

Set     Items   Description
S1     514506   DAG OR DIGRAPH OR GRAPH? ? OR DEPENDENCY OR CALL()(DIAGRAM?
              ? OR GRAPH? ? OR CHART? ?) OR TOPOLOGICAL()ORDER???
S2     208600   (CALL??? OR INVOK??? OR INVOCATION)(7N)(PROGRAM? ? OR OBJE-
             CT? ? OR CODE OR CLASS OR CLASSES OR MODULE? ? OR APPLICATION?
              ? OR SOFTWARE OR FIRMWARE OR FILE? ? OR METHOD? ? OR ROUTINE?
              ? OR SUBROUTINE? ? OR THREAD? ? OR PROCESS OR PROCESSES)
S3      87925   (PASS OR PASSES OR PASSING OR PASSAGE OR PASSED OR STAMP???
             OR COUPL??? OR RECEIV???)(5N)(ARGUMENT? ? OR PARAMETER? ? OR
             VARIABLE? ? OR METADATA OR META()DATA OR SIGNATURE? ? OR INTE-
             RFACE? ?)
S4    3919366   OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION? ? OR OPTIMISATIO-
             N? ? OR FINETUN??? OR FINE()TUN??? OR ENHANC??????
S5     524178   (INCREAS??? OR AUGMENT? OR BOOST? OR AMPLIF? OR RAIS??? OR
             IMPROV??? OR RAMP??? OR UPGRAD?)(5N)(EFFICIENC??? OR EFFICIENT
              OR EFFECTIV? OR SPEED OR EXECUT???)
S6    1933007   COST? ?
S7       1688   S1 AND S2 AND S4:S5
S8         12   S3 AND S7
S9     313765   (NONVOLATILE OR NON()VOLATILE)(3N)(STOR? OR MEMOR??? OR CI-
             RCUIT? OR AREA? ? OR PART? ? OR ELEMENT? ?) OR FIRMWARE OR ROM
              OR READ()ONLY()MEMORY OR PROM OR EPROM OR EEPROM OR FLASH()M-
             EMORY
S10         3   S7 AND S9
S11       610   S1 AND S2 AND S6
S12         0   S11 AND S3
S13         0   S11 AND S9
S14   6595213   EFFICIENC??? OR EFFICIENT OR EFFECTIV? OR SPEED
S15   1402728   (S4 OR S14)(7N)(PROGRAM? ? OR OBJECT? ? OR CODE OR CLASS OR
              CLASSES OR MODULE? ? OR APPLICATION? ? OR SOFTWARE OR FIRMWA-
             RE OR FILE? ? OR METHOD? ? OR ROUTINE? ? OR SUBROUTINE? ? OR -
             THREAD? ? OR PROCESSES)
S16      1491   S1 AND S2 AND S15
S17       964   S1 AND S3
S18        44   S17 AND S2
S19         0   S17 AND S9
S20        77   S17 AND S15
S21       113   S8 OR S10 OR S18:S20
```

```
S22          79   RD  (unique items)
S23          68   S22 NOT PY=2004:2007
S24           2   RD S10  (unique items)
S25        8912   EMBEDDED()(CODE OR PROGRAM? ? OR SOFTWARE OR APPLICATION? -
                  ?)
S26       29052   S1 AND S15
S27         126   S26 AND S25
S28          10   S26 AND FIRMWARE
S29         136   S27:S28
S30          76   RD  (unique items)
S31          46   S30 NOT (S23 OR PY=2004:2007)
```

23/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

10422439 E.I. No: EIP05229125752
Title: Cluster assignment of global values for clustered VLIW processors
Author: Terechko, Andrei; Le Thenaff, Erwan; Corporaal, Henk
Corporate Source: Philips Research, 5656AA Eindhoven, Netherlands
Conference Title: CASES 2003: International Conference on Compilers, Architecture, and Synthesis for Embedded Systems
Conference Location: San Jose, CA, United States Conference Date: 20031030-20031101
Sponsor: ACM SIGMICRO; IEEE TCuARCH
E.I. Conference No.: 64732
Source: CASES 2003: International Conference on Compilers, Architecture, and Synthesis for Embedded Systems CASES 2003: International Conference on Compilers, Architecture, and Synthesis for Embedded Systems 2003.
Publication Year: 2003
ISBN: 1581136765
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical)
Journal Announcement: 0506W2
Abstract: In this paper high-level language (HLL) variables that are alive in a whole HLL function, across multiple scheduling units, are termed as global values. Due to their long live ranges and, hence, large impact on the schedule, the global values require different compiler optimizations than local values, which span across only one scheduling unit. The instruction scheduler for a clustered ILP processor, which is responsible for cluster assignment of operations and variables, faces a difficult problem of assigning global values to clusters. Our study shows that trivial assignments (e.g. mapping all global values into one cluster) may result in a severe cycle count overhead relative to the unicluster of up to 26.3% for a four cluster VLIW machine. This paper presents three advanced algorithms for assigning global values to clusters based on multi- **pass** scheduling and affinity of **variables** . Furthermore, we measure performance of these algorithms on **optimized** multimedia C **applications** and assess quality of our algorithms by comparing them to a practical higher performance bound derived from a vast random search. Our algorithms reduce the execution time overhead of the best simple algorithm round-robin from 10.5% to 5.9% for the two cluster VLIW machine and from 17.3% to 14.12% for the four cluster VLIW machine. Copyright 2003 ACM. 32 Refs.
Descriptors: *Program processors; Program compilers; C (programming language); Feedback; Scheduling; Heuristic **methods** ; Random **processes** ; **Graph** theory; **Optimization** ; Algorithms
Identifiers: Instruction level parallelism (ILP); VLIW; Instruction schedulers; Cluster assignment; Register allocation
Classification Codes:
723.1.1 (Computer Programming Languages)
723.1 (Computer Programming); 731.1 (Control Systems); 912.2 (Management); 922.1 (Probability Theory); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 921.5 (Optimization Techniques)
723 (Computer Software, Data Handling & Applications); 731 (Automatic Control Principles & Applications); 912 (Industrial Engineering & Management); 921 (Applied Mathematics); 922 (Statistical Methods)
72 (COMPUTERS & DATA PROCESSING); 73 (CONTROL ENGINEERING); 91 (ENGINEERING MANAGEMENT); 92 (ENGINEERING MATHEMATICS)

23/5/2 (Item 2 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

10006075 E.I. No: EIP04368343280

Title: Post-pass compaction techniques
Author: De Bus, Bruno; Kastner, Daniel; Chanet, Dominique; Van Put, Ludo; De Sutter, Bjorn
Corporate Source: Electronics Systems Department Ghent University, Ghent, Belgium
Source: Communications of the ACM v 46 n 8 August 2003. p 41-46
Publication Year: 2003
CODEN: CACMA2 ISSN: 0001-0782
Language: English
Document Type: JA; (Journal Article) Treatment: G; (General Review)
Journal Announcement: 0409W2
Abstract: The techniques used in post-pass compaction tools that solve code-size-related programs in program development environments are discussed. The tools are aiPop, Diablo and squeeze++. The techniques consists of whole- **program** **optimizations** , elimination of duplicate **code** fragments in **program** , and the construction of an internal program representation. The results of whole- **program** analysis are used during post- **pass** **optimization** to remove unnecessary **parameter - passing** **code** and redundant register saving **code** that results from conservative adherence to **calling** conventions. Additionally, post-pass **optimizers** reapply standard compiler **optimizations** , such as peephole **optimization** , copy propagation, useless **code** elimination and strength reduction. (Edited abstract) 10 Refs.
Descriptors: *Data compression; Binary codes; Utility programs; C (programming language); **Object** oriented programming; **Program** compilers; **Graph** theory; **Optimization**
Identifiers: **Code** compaction; Compiler **optimization** ; Linkers; Post-pass compaction tools
Classification Codes:
723.1.1 (Computer Programming Languages)
723.2 (Data Processing); 723.1 (Computer Programming); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 921.5 (Optimization Techniques)
723 (Computer Software, Data Handling & Applications); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

23/5/4 (Item 4 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

09939457 E.I. No: EIP04298264484
Title: Heuristic algorithm based on a genetic algorithm for mapping parallel programs on hypercube multiprocessors
Author: Aguilar, Jose
Corporate Source: CEMISID Dpto. de Computacion Universidad de los Andes, Merida, Edo. Merida, Venezuela
Source: Computer Systems Science and Engineering v 18 n 4 July 2003. p 217-222
Publication Year: 2003
CODEN: CSSEEI ISSN: 0267-6192
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 0407W3
Abstract: In this work, we propose a heuristic algorithm based on Genetic Algorithm for the task-to-processor mapping problem in the context of local-memory multiprocessors with a hypercube interconnection topology. Hypercube multiprocessors have offered a cost effective and feasible approach to supercomputing through parallelism at the processor level by directly connecting a large number of low-cost processors with local memory which communicate by message **passing** instead of shared **variables** . We use concepts of the **graph** theory (task **graph** precedence to represent parallel programs, **graph** partitioning to solve the program decomposition problem, etc.) to model the problem. This problem is

NP-complete which means heuristic approaches must be adopted. We develop a heuristic algorithm based on Genetic Algorithms to solve it. 9 Refs.
Descriptors: *Multiprocessing **programs** ; Genetic algorithms; Storage allocation (computer); Cost **effectiveness** ; Problem solving; Heuristic **methods** ; Probability distributions; **Graph** theory; Topology; Mathematical models; Computer simulation
Identifiers: Heuristic algorithm; Hypercube multiprocessors; Parallel programs; Processor allocation
Classification Codes:
723.1 (Computer Programming); 722.1 (Data Storage, Equipment & Techniques); 911.2 (Industrial Economics); 723.4 (Artificial Intelligence); 922.1 (Probability Theory); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 723.5 (Computer Applications)
723 (Computer Software, Data Handling & Applications); 722 (Computer Hardware); 911 (Cost & Value Engineering; Industrial Economics); 921 (Applied Mathematics); 922 (Statistical Methods)
72 (COMPUTERS & DATA PROCESSING); 91 (ENGINEERING MANAGEMENT); 92 (ENGINEERING MATHEMATICS)

23/5/5 (Item 5 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

09789678 E.I. No: EIP04148097755
Title: Forward Dynamic Interprocedural Program Slicing
Author: Song, Yeong-Tae; Huynh, Dung T.
Corporate Source: Computer Science Program University of Texas at Dallas, Richardson, TX 75083, United States
Conference Title: 4th International Conference on Computer Science and Informatics, JCIS 1998
Conference Location: Research Triangle Park, NC, United States Conference Date: 19981023-19981028
Sponsor: Association for Intelligent Machinery; Duke University; Elsevier Publishing Company; Information Sciences Journal; US Army Research Office, Research Triangle Park, NC, USA
E.I. Conference No.: 62548
Source: Proceedings of the Joint Conference on Information Sciences v 3 1998.
Publication Year: 1998
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical)
Journal Announcement: 0404W1
Abstract: In this paper we present a forward dynamic program slicing algorithm for the interprocedural case. It is a recursive technique of analyzing **programs** that involve parameter passings and function **calls** (including recursive function **calls**). The analysis of two major **parameter passing methods** - **call** by value and call by reference and the effect of modification of global variables are discussed. For each function call the algorithm performs its main procedure which is based on Korel and Yalamanchili's algorithm and returns a function slice. Our algorithm is an **enhancement** of Korel and Yalamanchill's algorithm. We preserve the nodeslice of nonsimple blocks and use it only for the relevant slices of the variables instead of applying it to all the slices in a program and remove it from all the irrelevant slices afterwards. In order to compute dynamic slices for interprocedural programs correctly, we also consider return statements in our algorithm. 9 Refs.
Descriptors: *Computer programming; Dynamics; **Graph** theory; Computation theory; Algorithms; Set theory
Identifiers: Forward dynamic program; Interprocedural programs
Classification Codes:
723.1 (Computer Programming); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 721.1 (Computer Theory (Includes Formal Logic, Automata Theory, Switching Theory & Programming Theory))

723 (Computer Software, Data Handling & Applications); 921 (Applied Mathematics); 721 (Computer Circuits & Logic Elements)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

23/5/7 (Item 7 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

08998848 E.I. No: EIP02056845625
Title: Component identification method with coupling and cohesion
Author: Lee, Jong Kook; Jung, Seung Jae; Kim, Soo Dong; Jang, Woo Hyun; Ham, Dong Han
Corporate Source: Department of Computer Science Soongsil University, Dongjak-gu, Seoul, South Korea
Conference Title: 8th Asia Pacific Software Engineering Conference APSEC'2001
Conference Location: Macao, China Conference Date: 20011204-20011207
Sponsor: National Natural Science Foundation of China; Public Administration and Civil Service Bureau of Macau SAR; Companhia de Telecomunicacoes de Macau S.A.R.L.; Macau SAR Government Tourist Office
E.I. Conference No.: 58962
Source: Proceedings of the Asia-Pacific Software Engineering Conference and International Computer Science Conference, APSEC and ICSC 2001. p 79-86
Publication Year: 2001
Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications)
Journal Announcement: 0202W1
Abstract: Since the introduction of component-based development (CBD), effective component identification technique is known to be an important factor for successful CBD projects. As in CORBA Component Model by OMG, a component consists of one or more related objects, carrying out a homogeneous functionality. Most of the CBD methodologies utilize UML as the basic notational convention. Especially the component diagram or its variation is used to depict components. However, current CBD methodologies largely lack of systematic component identification algorithm that can be **effectively** used to group related use-cases and **classes** into components. In this paper, we introduce component identification method that considers component **coupling** , cohesion, **dependency** , **interface** , granularity, and architecture. We also provide a case study on a large-scaled real CBD project, in which the proposed method was applied. 10 Refs.
Descriptors: *Software engineering; Object oriented programming; Algorithms; Interfaces (computer); Computer architecture; Computer software maintenance; Computer software reusability; Electronic commerce
Identifiers: Component based software engineering; Component identification; Common object request broker architecture; Object management group
Classification Codes:
723.1 (Computer Programming); 722.2 (Computer Peripheral Equipment); 723.5 (Computer Applications)
723 (Computer Software, Data Handling & Applications); 722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING)

23/5/10 (Item 10 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

07256306 E.I. No: EIP95092862210
Title: Graph **-based optimizations for** parameter passing **in remote invocations**
Author: Lopes, Cristina Videira
Corporate Source: Northeastern Univ, Boston, MA, USA

Conference Title: Proceedings of the 4th International Workshop on Object Orientation in Operating Systems
Conference Location: Lund, Swed Conference Date: 19950814-19950815
Sponsor: IEEE
E.I. Conference No.: 43594
Source: International Workshop on Object Orientation in Operating Systems - Proceedings 1995. IEEE, Los Alamitos, CA, USA,94TB8120. p 179-182
Publication Year: 1995
CODEN: 002130 ISSN: 1063-5351
Language: English
Document Type: CA; (Conference Article) Treatment: G; (General Review); T; (Theoretical)
Journal Announcement: 9511W3
Abstract: This paper proposes a **graph** -based solution to the problem of **parameter passing** in distributed object-oriented applications. The solution presented here makes use of **graph** directives for the purpose of **optimizing** the copying of **objects** which are **passed** as **parameters** in remote invocations. These specifications are done using a very simple **graph** traversal language - GOOP. The resulting **parameter passing** scheme can be drastically improved for every situation in the applications. (Author abstract) 10 Refs.
Descriptors: ***Object** oriented programming; **Optimization** ; **Graph theory** ; Specifications; Computer programming languages; Computer operating systems; Distributed computer systems; Algorithms
Identifiers: **Parameter passing** ; Remote invocations; Software Package GOOP; **Graph** directives
Classification Codes:
723.1.1 (Computer Programming Languages)
723.1 (Computer Programming); 921.5 (Optimization Techniques); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 902.2 (Codes & Standards)
723 (Computer Software); 921 (Applied Mathematics); 902 (Engineering Graphics & Standards)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS); 90 (GENERAL ENGINEERING)

23/5/11 (Item 11 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

07172118 E.I. No: EIP95052724543
Title: New programming technique for lazy functional languages
Author: van Gilst, F.A.; van den Broek, P.M.
Corporate Source: Univ of Twente, Enschede, Neth
Source: Science of Computer Programming v 24 n 1 Feb 1995. p 63-81
Publication Year: 1995
CODEN: SCPGD4 ISSN: 0167-6423
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 9507W4
Abstract: In this paper we present a new programming technique for lazy functional programming languages. The technique is embedded in a programming methodology which is based on divide and conquer: the division of problems into subproblems. Such a division will be represented by a **call graph** . A class of **program** schemes, which implement **call graphs** , is derived based on Johnsson's approach to attribute evaluation in attribute grammars. The key idea is to consider the **passing** of **arguments** to functions and the return of results by functions in a **call graph** as the propagation of inherited and synthesized attributes, respectively, in an attribute grammar. The new technique is illustrated by several small examples and a case study: the design and implementation of a compiler generator. (Author abstract) 9 Refs.
Descriptors: *Computer programming; Computer programming languages;

Programming theory; Computational grammars; Functions; Program compilers; Software engineering; Formal logic; Problem solving
Identifiers: Lazy functional programming languages; **Call graphs** ; Compiler generator
Classification Codes:
723.1.1 (Computer Programming Languages)
723.1 (Computer Programming); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory)
723 (Computer Software); 721 (Computer Circuits & Logic Elements)
72 (COMPUTERS & DATA PROCESSING)

23/5/13 (Item 13 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

06875957 E.I. No: EIP94061317585
Title: MOSAda: methode et outils de simulation en Ada
Title: MOSada: methodology and tools for simulations in Ada
Author: Bourgeois, R.; Demailly, G.
Corporate Source: THOMSON-CSF
Source: Revue Technique Thomson - CSF v 25 n 2 Jun 1993. p 533-547
Publication Year: 1993
CODEN: RTTCBG ISSN: 0035-4279
Language: French
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 9407W4
Abstract: MOSAda is the simulation environment dedicated to Ada realized by the Electronics Systems Division of THOMSON-CSF to **enhance** the development of its products, from the faisability and validation studies to the development of **Software** Configuration Components (also **called** Operational **Programs**). The use of MOSAda, which shares with these Operational Programs the language Ada and libraries organized as reusable software components (for mathematics or abstract data types), increase reusability, portability, and competences. The MOSAda development process is completely included in the THOMSON CASE tool (which consists on a customization of the Software To Pictures tool developed by IDE) and methodology. The first step, which ends with an analysis specification uses the SA/RT methodology where a system is discribed with processes, data, and controls. A process modifies continuously or discretely some input data to output, and the process scheduling is based on state diagrams where events are controls combinations. Simulation designers find the event concept in the MOSAda design methodology. This methodology, based on Object Oriented Design, proposes to define active or passive objects (which model systems of the real world) and the MOSAda tools provide a simulation kernel to manage events for these objects. Active objects have a separated thread of control (represented in Ada by a task) and are activated each time they receive an event, and passive objects may be activated only by the active ones. For continuous changes of states. MOSAda provides integrated active objects with automatic processing of the continuous state **variables** (these objects may be **coupled** or not). Decoupling different **classes** of **objects** is **enhanced** by the use of properties where communications (based on client server model) between two objects are performed (as an example, a radar class should not use an airplane class to perform detection, an electromagnetic waves property should interface these classes). The different steps of a MOSAda design phase are identifying the objects (distinguishing active or passive objects and listing the related events), identifying the properties (with their client server object classes), building a design **dependency graph** , determining the behavior of objects (actions performed on particular events, analytical or differential equations for continuous variables), determining data flows between different classes and drawing them (Buhr diagrams). Code generation is simplified by the use of template files available for all the packages to be developed. MOSAda provides a man machine interface (based on X-Windows

and Motif) for the control of a simulation. This interface is decoupled from the simulation (to **enhance** readability and development time of **objects**) and all simulation objects automatically send it their state (which can be edited or used to display graphics). Another textual interface is provided to perform the basic control functions on a simulation (step, go, set break, quit, list events). MOSAda objects may generate ASCII or binary files containing some of their internal state variables for post processing exploitation. Statistics may also be generated. (Author abstract) 9 Refs.
Descriptors: *Computer simulation languages; Computer software; Ada (programming language); Computer aided software engineering; Database systems; User interfaces; State assignment; Equivalence classes; Computer simulation; Computer software portability
Identifiers: MOSAda software package; Software configuration components; State diagrams; Code generation; State variables; Data flows; Process scheduling; Computer hardware specification languages
Classification Codes:
723.1.1 (Computer Programming Languages)
723.1 (Computer Programming); 723.5 (Computer Applications); 723.3 (Database Systems); 722.2 (Computer Peripheral Equipment); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory)
723 (Computer Software); 722 (Computer Hardware); 721 (Computer Circuits & Logic Elements)
72 (COMPUTERS & DATA PROCESSING)

23/5/14 (Item 14 from file: 8)

06787911 E.I. No: EIP94011187509
Title: Optimal reduction in weak- lambda -calculus with shared environments
Author: Yoshida, Nobuko
Corporate Source: Keio Univ, Yokohama, Jpn
Conference Title: Proceedings of the 6th International Conference on Functional Programming Languages and Computer Architecture (FPCA '93)
Conference Location: Copenhagen, Den Conference Date: 19930609-19930611
Sponsor: ACM SIGPLAN/SIGARCH
E.I. Conference No.: 19830
Source: Proc 6 Int Conf Funct Program Lang Comput Archit 1993. Publ by ACM, New York, NY, USA. p 243-251
Publication Year: 1993
ISBN: 0-89791-595-X
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical)
Journal Announcement: 9403W1
Abstract: We introduce a weak- lambda -calculus called lambda f-calculus which formalizes functional execution with shared environments in a clean and tractable way. It effectively incorporates the notion of name **passing** into the functional regime, treating **variables** in environments as channels of communication. This enables simple formulation of reduction rules under a sequence of shared environments, resulting in a functional calculus whose weak reduction enjoys several pleasant syntactic properties, e.g. the Church-Rosser property and normalizability. We also show that the leftmost reduction strategy of the calculus is optimal in the weak execution scheme. (Author abstract) 12 Refs.
Descriptors: *Formal logic; Automata theory; Computer **software** ; Errors; **Optimization** ; **Graph theory** ; Theorem proving; Optimal systems
Identifiers: Shared environments; Lambda calculus; Church Rosser property ; Normalizability
Classification Codes:
721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 723.1 (Computer Programming); 921.5

(Optimization Techniques); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory)
721 (Computer Circuits & Logic Elements); 723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

23/5/15 (Item 15 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

06694679 E.I. No: EIP93081063632
Title: Efficient call graph **analysis**
Author: Hall, Mary W.; Kennedy, Ken
Corporate Source: Stanford Univ, Stanford, CA, USA
Source: ACM Letters on Programming Languages and Systems v 1 n 3 Sep 1992. p 227-242
Publication Year: 1992
CODEN: ALPSE8 ISSN: 1057-4514
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9310W4
Abstract: We present an efficient algorithm for computing the procedure **call graph** , the **program** representation underlying most interprocedural **optimization** techniques. The algorithm computes the possible bindings of procedure variables in languages where such variables only **receive** their values through **parameter passing** , such as Fortran. We extend the algorithm to accommodate a limited form of assignments to procedure variables. The resulting algorithm can also be used in analysis of functional programs that have been converted to Continuation-Passing Style. We discuss the algorithm in relationship to other **call graph** analysis approaches. Many less efficient techniques produce essentially the same **call graph** . A few algorithms are more precise, but they may be prohibitively expensive depending on language features. (Author abstract) 24 Refs.
Descriptors: *Algorithms; **Graph** theory; **Optimization** ; **Program compilers** ; Computer programming languages; **Subroutines** ; Function evaluation; Computational methods; Data reduction
Identifiers: Procedure **call graph** ; Interprocedural dataflow analysis
Classification Codes:
723.1.1 (Computer Programming Languages)
723.1 (Computer Programming); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 921.5 (Optimization Techniques); 723.2 (Data Processing)
723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

23/5/16 (Item 16 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

06030040 E.I. Monthly No: EI9103024818
Title: Data flow analysis of distributed communicating processes.
Author: Reif, John H.; Smolka, Scott A.
Corporate Source: Duke Univ, Durham, NC, USA
Source: International Journal of Parallel Programming v 19 n 1 Feb 1990 p 1-30
Publication Year: 1990
CODEN: IJPPE5 ISSN: 0885-7458
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9103
Abstract: Data flow analysis is a technique essential to the compile-time **optimization** of computer **programs** , wherein facts relevant to **program**

optimizations are discovered by the global propagation of facts obvious locally. This paper extends several known techniques for data flow analysis of sequential programs to the static analysis of distributed communicating processes. In particular, we present iterative algorithms for detecting unreachable program statements, and for determining the values of program expressions. The latter information can be used to place bounds on the size of variables and messages. Our main innovation is the event spanning **graph**, which serves as a heuristic for ordering the nodes through which data flow information is propagated. We consider both static communication, where all channel arguments are constants, and the more difficult dynamic communication, where all channel arguments may be **variables** and channels may be **passed** as messages. (Author abstract) 22 Refs.

Descriptors: *COMPUTER PROGRAMMING--*Optimization; MATHEMATICAL TECHNIQUES--Iterative Methods; COMPUTER SYSTEMS, DIGITAL--Distributed; DATA PROCESSING--Data Transfer; MATHEMATICAL TECHNIQUES-- **Graph** Theory

Identifiers: DATA FLOW ANALYSIS; DISTRIBUTED COMMUNICATING PROCESSES; MESSAGE PASSING

Classification Codes:
723 (Computer Software); 921 (Applied Mathematics); 722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

23/5/17 (Item 17 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

05792296 E.I. Monthly No: EIM8909-030653
Title: Fast interprocedural alias analysis.
Author: Cooper, Keith D.; Kennedy, Ken
Corporate Source: Rice Univ, Houston, TX, USA
Conference Title: Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages
Conference Location: Austin, TX, USA Conference Date: 19890111
Sponsor: ACM; Special Interest Group on Automata and Computability Theory ; Special Interest Group on Programming Languages
E.I. Conference No.: 12326
Source: Conf Rec Sixteenth Annu ACM Symp Princ Program Lang. Publ by ACM, New York, NY, USA. p 49-59
Publication Year: 1989
ISBN: 0-89791-294-2
Language: English
Document Type: PA; (Conference Paper) Treatment: T; (Theoretical)
Journal Announcement: 8909
Abstract: We present a new algorithm for computing interprocedural aliases due to **passing parameters** by reference. This algorithm runs in O(N**2 plus NE) time and, when combined with algorithms for alias-free, flow-insensitive data-flow problems, yields algorithms for solution of the general flow-insensitive problems that also run in O(N**2 plus NE) time. (Author abstract) 18 Refs.
Descriptors: *COMPUTER PROGRAMMING--*Algorithms; MATHEMATICAL TECHNIQUES -- **Graph** Theory
Identifiers: INTERPROCEDURAL ALIASES; **SUBROUTINE** INVOCATION; DATA FLOW PROBLEMS; **CALL GRAPHS**
Classification Codes:
723 (Computer Software); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

23/5/19 (Item 19 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04489475 E.I. Monthly No: EI8402010562 E.I. Yearly No: EI84023608
Title: OPTIMIZING PROCEDURE CALLS AND RETURNS.

Author: Er, M. C.
Corporate Source: Univ of Wollongong, Dep of Computing Science, Wollongong, NSW, Aust
Source: Software - Practice and Experience v 13 n 10 Oct 1983 p 921-939
Publication Year: 1983
CODEN: SPEXBL ISSN: 0038-0644
Language: ENGLISH
Journal Announcement: 8402
Abstract: When a procedure is activated as a logical last statement in another procedure, optimization could be done to the procedure call and return. Several optimizations are discussed. The first optimization is based on the idea of reusing the current stack frame. With this approach, it is necessary to compute the order of **passing parameters** . A tactic of computing the **dependency** of parameters and a heuristic method of breaking the cyclic dependencies are therefore proposed. However, this approach has severe restrictions imposed on the kinds of procedure that can be called; in particular, the called procedure cannot contain references to the calling procedure. To remove all such restrictions, the second and third **methods** of **optimization** are put forwards. The second **optimization** revolves around the concept of continuation closure. It uses the normal stack frame of standard implementation, but alters its continuation closure to achieve quick procedure exit. The third optimization builds upon the concept of para-stack frame and common continuation closure for achieving fast procedure exit. Various other refinements are also considered. 5 refs.
Descriptors: *COMPUTER PROGRAMMING LANGUAGES
Classification Codes:
723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

23/5/21 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01474229 ORDER NO: AADAA-I9610479
PARTITIONING DEPENDENCY GRAPHS **FOR CONCURRENT EXECUTION: A PARALLEL SPREADSHEET ON A REALISTICALLY MODELED MESSAGE PASSING ENVIRONMENT**
Author: WACK, ANDREW P.
Degree: PH.D.
Year: 1995
Corporate Source/Institution: UNIVERSITY OF DELAWARE (0060)
Professor In Charge: B. DAVID SAUNDERS
Source: VOLUME 56/12-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 6876. 155 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

Effective partitioning of **programs** for parallel processing has become very important. We propose using spreadsheets as a convenient model and tool for parallel computation. We wish to create an easy way to exploit an existing parallel processing machine available to many people: the network of workstations. Using results from experiments with C-Linda on a network of workstations, we have developed a novel machine model that includes parameters representing the high overhead of starting a communication experienced in a message **passing** system. We use five **parameters** to characterize the cost of sending a message between two processors; communication start up time for the sender, time the sender takes to send each byte in the message, corresponding **parameters** for the **receiver** , and the transmission time per byte through the communication channel. A formula is presented for predicting these parameters under C-Linda, given workstation and network speed. A unique feature of this model is that it captures the time when the workstation is occupied during a communication event.

We approach the problem of parallelizing spreadsheet programs as a matter of partitioning their **dependency graphs** . In this structure,

nodes of the **graph** represent computations to be performed and the edges between the nodes represent the data dependencies between the computations.

We consider the problem of finding an optimum partition of the **dependency graph** given the above model of communication. We examine various restrictions on the **dependency graph** , node weights, edge weights, and number of processors. Unlike most other work on this type of problem, ours takes into account the possible gain achieved when two or more communications are combined, thus eliminating some of the start up costs (latencies).

We give a linear time algorithm for finding an optimum partition for a very restricted class of **dependency graphs** , which, however, includes many important cases. We further show that in many instances relaxing any one of the restrictions makes the problem NP-complete. We discuss heuristic extensions to our algorithm and show simulation results that demonstrate the superior performance of our algorithm over current heuristics.

23/5/22 (Item 3 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01151751 ORDER NO: AAD91-09517
A MODEL FOR FINE-GRAINED ASYNCHRONOUS CONCURRENCY THROUGH PARALLEL GRAPH **REDUCTION**
Author: SCHAEFER, BARTON E.
Degree: PH.D.
Year: 1990
Corporate Source/Institution: OREGON GRADUATE INSTITUTE OF SCIENCE & TECHNOLOGY (0284)
Supervisor: RICHARD B. KIEBURTZ
Source: VOLUME 51/11-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 5434. 219 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

This thesis explores techniques for massively parallel computation on MIMD computers executing fine-grained computational tasks asynchronously. It presents a model for evaluating expressions by concurrent **graph** reduction. The nodes of a computation **graph** are represented in the memories of a network of identical computing modules. The thesis presents experimental studies of the behavior of a dynamic scheduling algorithm for distributing workload over the **modules** of a network. **Called** diffusion scheduling, it uses a measure of workload as the analog of pressure to direct tasks to modules where they are most likely to receive prompt service. A second series of experiments investigates the effectiveness of speculative evaluation in stimulating concurrent activity when the more commonly employed approaches of data or control parallelism fail. **Parameters** of network dimension, message **passing** characteristics, and data dependencies within programs are considered in development of a heuristic method for creating and distributing speculative work.

23/5/23 (Item 4 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01135458 ORDER NO: AAD90-35159
AN OPERAND DEPENDENCE GRAPH METHOD **FOR** CODE OPTIMIZATION
Author: ASURU, JONATHAN M.
Degree: PH.D.
Year: 1990
Corporate Source/Institution: OKLAHOMA STATE UNIVERSITY (0664)
Source: VOLUME 51/07-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 3449. 175 PAGES
Descriptors: COMPUTER SCIENCE

Descriptor Codes: 0984

Scope and method of study. An approach for improving the intermediate code of **programs** with structured control flow **graphs** in one **optimization** pass is developed. Control structure analysis is performed first to define forward reachability, predominance, and post-dominance relations on the nodes of a program flow **graph** and to partition a **program** flow **graph** into **optimization** regions. Then intraprocedural alias information of a **program** is obtained by applying an alias analysis procedure. **Code optimization** procedures are applied to each **program** region according to some **topological order** . A program region is represented with an operand dependence **graph** and each distinct instance of a variable is assigned a unique version number and propagated along forward control flow paths to variable use points. Each statement instance is **stamped** with an instance **signature** (n-tuple of reaching source operands version numbers). The instance signatures of statements is used to expose potential redundancies among lexically identical expressions. A concept of path cover is used to determine when potentially redundant expressions are fully redundant.

Findings and conclusions. A region based code improvement method is adequate to detect most of the feasible common subexpression, **code** hoisting, and **code** sinking **optimizations** as well as all the feasible loop **optimizations** in a single **code optimization** pass. The operand dependence **graph** structure and the path cover concept present a uniform framework for recognizing redundancies involving lexically identical expressions. The worst case time complexity for redundant statements detection is O(r $\star$ π), where r is the total number of potentially redundant statements and π is the data flow width of a program flow **graph** . Simple linear recurrence array references in a program loop can be replaced with scalar copy statements to improve loop execution time.

23/5/25 (Item 2 from file: 2)

08947704 INSPEC Abstract Number: C2004-06-6150N-021

Title: Compiler optimized **remote** method invocation

Author(s): Veldema, R.; Philippsen, M.
Author Affiliation: Comput. Sci. Dept., Elargen-Nuremberg Univ., Erlangen, Germany

Conference Title: Proceedings. IEEE International Conference on Cluster Computing p.127-36
Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA
Publication Date: 2003 Country of Publication: USA xxii+520 pp.
ISBN: 0 7695 2066 9 Material Identity Number: XX-2004-00104
U.S. Copyright Clearance Center Code: 0 7695 2066 9/2003/$17.00
Conference Title: Proceedings. IEEE International Conference on Cluster Computing
Conference Sponsor: IEEE; IEEE Comput. Soc.; Univ. of Hong Kong; Hong Kong Section Comput. Chapter; Comput. Soc. Task Force on Cluster Computing
Conference Date: 1-4 Dec. 2003 Conference Location: Hong Kong, China
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)

Abstract: We further **increase** the **efficiency** of Java RMI **programs** . Where other **optimizing** re-implementations of RMI use pre-processors to create stubs and skeletons and to create class specific serializers and deserializers, this paper demonstrates that with transformations based on compile time analysis, an additional 18% performance gain can be achieved over class specific serializers alone for a simple scientific application. A novel and RMI-specific version of static heap analysis is used to derive information about objects that are **passed** as **arguments** of remote method invocations. This knowledge of **objects** and their interrelations is used for three **optimizations** . First, dynamic introspection and/or (recursive) dynamic invocations of object specific serializers is slow. With knowledge

from our heap analysis, the marshaling of **graphs** of argument **objects** can be inlined at the **call** site. Hence, many **method** table lookups and skeleton indirections of previous approaches can be avoided and less protocol information is sent over the network. Secondly, because object **graphs** may be **passed** as RMI **arguments** , cyclic references need to be detected. With our heap analysis, we can detect if there is no potential for cycles and hence, cycle detection code can be left out of the serialization and marshaling codes. Finally, object arguments to remote methods cause object creation and garbage collection. Heap analysis and an RMI-specific version of escape analysis allows the reuse of object **graphs** created in earlier remote invocations. (19 Refs)
Subfile: C
Descriptors: Java; **optimising** compilers; remote procedure calls
Identifiers: compiler **optimized** remote **method invocation** ; Java RMI program; preprocessors; class specific serializers; class specific deserializers; compile time analysis; RMI-specific version; static heap analysis; dynamic introspection; recursive dynamic invocations; object specific serializers; table lookups; skeleton indirections; protocol; object **graphs** ; RMI argument; cyclic references; cycle detection code; marshaling codes; object argument; object creation; garbage collection; escape analysis
Class Codes: C6150N (Distributed systems software); C6150C (Compilers, interpreters and other processors); C6110J (Object-oriented programming)

23/5/30 (Item 7 from file: 2)
DIALOG(R)File 2:INSPEC

07936844 INSPEC Abstract Number: C2001-07-6110J-009
Title: Alias analysis method for object oriented programs using alias flow graphs
Author(s): Ohata, F.; Kondou, K.; Inoue, K.
Author Affiliation: Graduate Sch. of Eng. Sci., Osaka Univ., Japan
Journal: Transactions of the Institute of Electronics, Information and Communication Engineers D-I vol.J84D-I, no.5 p.443-53
Publisher: Inst. Electron. Inf. & Commun. Eng,
Publication Date: May 2001 Country of Publication: Japan
CODEN: DTRDES ISSN: 0915-1915
SICI: 0915-1915(200105)J84DI:5L.443:AAMO;1-O
Material Identity Number: M972-2001-006
Language: Japanese Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: When an expression possibly refers to a memory location which is referred to by another expression, we can say there is an alias relation between those expressions. Alias relations are generated by **parameter passing** , reference **variables** , indirect reference with pointer variables, and so on. Since existing alias analysis methods do not reuse analysis results previously done before, we were unable to expect **efficient** alias analysis for **object** oriented **programs** in which reusability is essential. We propose an alias analysis method for object oriented programs, which considers reusability and modularity of its results. We have implemented this method as a JAVA alias analysis tool. This tool can deal with large programs such as class libraries in JDK (JAVA Developer's Kit), and we show some experimental results of applying this tool to several large JAVA programs. (16 Refs)
Subfile: C
Descriptors: **graph** theory; Java; object-oriented programming; software libraries; software reusability; storage management
Identifiers: alias analysis method; object oriented programs; alias flow **graphs** ; memory location; alias relation; **parameter passing** ; reference variables; indirect reference; pointer variables; reusability; JAVA alias analysis tool; large programs; class libraries; JDK; JAVA Developers Kit; large JAVA programs

Class Codes: C6110J (Object-oriented programming); C6110B (Software engineering techniques); C6140D (High level languages); C6150N (Distributed systems software); C6120 (File organisation); C1160 (Combinatorial mathematics)

23/5/33 (Item 10 from file: 2)
DIALOG(R)File 2:INSPEC

07656493 INSPEC Abstract Number: B2000-09-2550-001, C2000-09-1230R-014
Title: Influence-based model decomposition
Author(s): Bailey-Kellogg, C.; Feng Zhao
Author Affiliation: Dartmouth Coll., Hanover, NH, USA
Conference Title: Proceedings Sixteenth National Conference on Artificial Intelligence (AAI-99). Eleventh Innovative Applications of Artificial Intelligence Conference (IAAI-99) p.402-9
Publisher: AAAI Press, Menlo Park, CA, USA
Publication Date: 1999 Country of Publication: USA xxvi+998 pp.
ISBN: 0 262 51106 1 Material Identity Number: XX-1999-01742
Conference Title: Proceedings Sixteenth National Conference on Artificial Intelligence (AAAI-99). Eleventh Innovative Applications of Artificial Intelligence Conference (IAAI-99)
Conference Sponsor: American Assoc. Artificial Intelligence; ACM/SIGART; Defense Advance Res. Projects Agency; et al
Conference Date: 18-22 July 1999 Conference Location: Orlando, FL, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Applications (A); Theoretical (T)
Abstract: Rapid advances in MEMS and information processing technology have enabled a new generation of AI robotic systems-so-called smart matter systems-that are sensor rich and physically embedded. These systems range from decentralized control systems that regulate building temperature (smart buildings) to vehicle on-board diagnostic and control systems that interrogate large amounts of sensor data. One of the core tasks in the construction and operation of these smart matter systems is to synthesize optimal control policies using data rich models for the systems and environment. Unfortunately, these models may contain thousands of **coupled** real-valued **variables** and are prohibitively expensive to reason about using traditional optimization techniques such as neural nets and genetic algorithms. The paper introduces a general mechanism for automatically decomposing a large model into smaller subparts so that these subparts can be separately optimized and then combined. The mechanism decomposes a model using an influence **graph** that records the coupling strengths among constituents of the model. The paper demonstrates the mechanism in an **application** of decentralized **optimization** for a temperature regulation problem. Performance data has shown that the approach is much more efficient than the standard discrete optimization algorithms and achieves comparable accuracy. (16 Refs)
Subfile: B C
Descriptors: common-sense reasoning; genetic algorithms; model-based reasoning; optimal control; rapid thermal processing; temperature control
Identifiers: influence-based model decomposition; AI robotic systems; smart matter systems; data rich models; influence **graph** ; decentralized optimization; temperature regulation problem
Class Codes: B2550 (Semiconductor device technology); C1230R (Reasoning and inference in AI); C6170K (Knowledge engineering techniques); C1330 (Optimal control); C1180 (Optimisation techniques); C3120N (Thermal variables control); C3350E (Control applications in the electronics industry)

23/5/34 (Item 11 from file: 2)
DIALOG(R)File 2:INSPEC

06838698 INSPEC Abstract Number: C9804-4240P-013
Title: Design and performance of parallel and distributed approximation algorithms for maxcut
Author(s): Homer, S.; Peinado, M.
Author Affiliation: Dept. of Comput. Sci., Boston Univ., MA, USA
Journal: Journal of Parallel and Distributed Computing vol.46, no.1 p.48-61
Publisher: Academic Press,
Publication Date: 10 Oct. 1997 Country of Publication: USA
CODEN: JPDCER ISSN: 0743-7315
SICI: 0743-7315(19971010)46:1L.48:DPPD;1-I
Material Identity Number: G544-98002
U.S. Copyright Clearance Center Code: 0743-7315/97/$25.00
Language: English Document Type: Journal Paper (JP)
Treatment: Applications (A); Practical (P)
Abstract: We develop and experiment with a new parallel algorithm to approximate the maximum weight cut in a weighted undirected **graph** . Our implementation starts with the recent (serial) algorithm of M.X. Goemans and D.P. Williamson (1995) for this problem. We consider several different versions of this algorithm, varying the interior-point part of the algorithm in order to **optimize** the parallel **efficiency** of our **method** . Our work aims for an **efficient** , practical formulation of the algorithm with close-to-optimal parallelization. We analyze our parallel algorithm in the LogP model and predict linear speedup for a wide range of the parameters. We have implemented the algorithm using the message **passing interface** (MPI) and run it on several parallel machines. In particular, we present performance measurements on the IBM SP2, the Connection Machine CM5, and a cluster of workstations. We observe that the measured speedups are predicted well by our analysis in the LogP model. Finally, we test our implementation on several large **graphs** (up to 13,000 vertices), particularly on large instances of the Ising model. (41 Refs)
Subfile: C
Descriptors: distributed algorithms; message passing; parallel algorithms ; performance evaluation
Identifiers: distributed approximation algorithms; maxcut; parallel approximation algorithms; performance; maximum weight cut; weighted undirected **graph** ; close-to-optimal parallelization; LogP model; message **passing interface** ; performance measurements; IBM SP2; Connection Machine CM5; cluster of workstations; Ising model
Class Codes: C4240P (Parallel programming and algorithm theory); C5440 (Multiprocessing systems); C5220P (Parallel architecture); C6150N (Distributed systems software); C5470 (Performance evaluation and testing)

23/5/35 (Item 12 from file: 2)
DIALOG(R)File 2:INSPEC

06796942 INSPEC Abstract Number: C9802-6150C-025
Title: Aliases and their effect on data dependency **analysis**
Author(s): Parimaladevi, R.; Subramanian, R.K.
Journal: Malaysian Journal of Computer Science vol.10, no.2 p.45-50
Publisher: Univ. Malaya,
Publication Date: Dec. 1997 Country of Publication: Malaysia
CODEN: MJCSFS ISSN: 0127-9084
SICI: 0127-9084(199712)10:2L.45:ATED;1-L
Material Identity Number: E386-97003
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: Parallelising compilers try to automatically convert sequential programs into parallel programs to be executed on the targeted parallel machine. The main task of the parallelising compiler is to locate the areas

of potential parallelism in the sequential programs. The major problem in doing so is the data **dependency** in the programs. These could be identified by one or more **passes** of the program if unique **variable** names are used for the memory locations. If different variable names are used to point to the same memory location it causes a different dimension to the problem. If different variable names refer to the same location they are called aliases. Aliases could occur when there is a subprogram call. The **parameters passed** to the subprogram could be aliases in the subprogram itself or be aliases to the variable used in the **calling program** . Aliases occur during the usage of recursive data structures. Parameter scoping could also lead to alias problem when a global **variable** is **passed** to the subprogram as a **parameter** . To handle the problem of aliases the compiler will have to perform a detailed alias analysis so that suitable parallel codes could be generated. The alias problem has been examined and methods of identifying the occurrences of aliases have been developed. The methods adopted by the tool for handling the aliases in C programs have been described. (11 Refs)

Subfile: C

Descriptors: data structures; parallel programming; parallelising compilers

Identifiers: data **dependency** analysis; parallelising compilers; sequential programs; parallel programs; targeted parallel machine; data **dependency** ; variable names; aliases; recursive data structures; parallel codes; C programs

Class Codes: C6150C (Compilers, interpreters and other processors); C6110P (Parallel programming); C6120 (File organisation)

23/5/37 (Item 14 from file: 2)

06276363 INSPEC Abstract Number: C9607-1340B-017

Title: Decoupling of MIMO systems by graph **-theoretic approach**

Author(s): Fanti, M.P.; Maione, B.; Turchiano, B.

Author Affiliation: Dipartimento di Elettrotecnica ed Electtronica, Politecnico di Bari, Italy

Journal: Studies in Informatics and Control vol.5, no.1 p.49-67

Publisher: Informatics & Control Publications,

Publication Date: March 1996 Country of Publication: Romania

CODEN: SICOF7 ISSN: 1220-1766

SICI: 1220-1766(199603)5:1L.49:DMSG;1-A

Material Identity Number: D226-96003

Language: English Document Type: Journal Paper (JP)

Treatment: Theoretical (T)

Abstract: Decoupling techniques based on state variable concepts are appreciated by control system theorists but the entries of the state, input and output matrices of the state-space representation are regarded as exactly known numerical values, while practising engineers have to cope with varying parameters and unavoidable uncertainties. In order to overcome these disadvantages of the space-state theory, Reinschke (1988) has proposed a **graph** -theoretic approach to the decoupling problem. The necessary and sufficient conditions for decoupling are directly interpreted in terms of properties of the **digraph** associated with the state-space equations. Using digraphs gives insight into the structural nature of the decoupling property, showing how the feedback coefficients offset the original **coupling** between the terminal **variables** of the plant. Unfortunately, in order to apply this design method, it is necessary that the **digraph** associated with the plant equations presents a particular structure. This paper overcomes the limitations of the Reinschke approach by transforming the plant equations into a properly chosen canonical form before associating the **digraph** with them. Any system which can be decoupled by a state feedback controller can be reduced to a canonical form which belongs to the class for which it is possible to apply the Reinschke

design method. The paper shows an application of the proposed approach to the synthesis of the controller of a synchronous machine supplying an infinite busbar. The example illustrates the **effectiveness** and ease of use of the **method** . (13 Refs)
Subfile: C
Descriptors: busbars; control system synthesis; directed **graphs** ; machine control; MIMO systems; state feedback; state-space methods; synchronous machines
Identifiers: MIMO systems decoupling; **graph** -theoretic approach; feedback coefficients; varying parameters; uncertainties; necessary conditions; sufficient conditions; directed **graph** ; state-space equations; plant terminal variables; Reinschke design method; canonical form; state feedback controller; controller synthesis; synchronous machine; infinite busbar
Class Codes: C1340B (Multivariable control systems); C1310 (Control system analysis and synthesis methods); C1160 (Combinatorial mathematics); C3340H (Control of electric power systems)

23/5/40 (Item 17 from file: 2)
DIALOG(R)File 2:INSPEC

04226744 INSPEC Abstract Number: C88058457
Title: Minimizing register usage penalty at procedure calls
Author(s): Chow, F.C.
Author Affiliation: MIPS Comput. Syst. Inc., Sunnyvale, CA, USA
Journal: SIGPLAN Notices vol.23, no.7 p.85-94
Publication Date: July 1988 Country of Publication: USA
CODEN: SINODQ ISSN: 0362-1340
U.S. Copyright Clearance Center Code: 0 89791 269 1/88/0006/0085$1.50
Conference Title: SIGPLAN '88 Conference on Programming Language Design and Implementation
Conference Sponsor: ACM
Conference Date: 22-24 June 1988 Conference Location: Atlanta, GA, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: Inter-procedural register allocation can minimize the register usage penalty at procedure calls by reducing the saving and restoring of registers at procedure boundaries. A one-pass inter-procedural register allocation scheme based on processing the procedures in a depth-first traversal of the **call graph** is presented. This scheme can be overlayed on top of intra-procedural register allocation via a simple extension to the priority-based coloring algorithm. Using two different usage conventions for the registers, the scheme can distribute register saves/restores throughout the **call graph** even in the presence of recursion, indirect calls, or separate compilation. A natural and efficient way to **pass parameters** emerges from this scheme. A separate technique uses data flow analysis to **optimize** the placement of the save/restore **code** for registers within individual procedures. The techniques described have been implemented in a production compiler suite. Measurements of the effects of these techniques on a set of practical programs are presented and the results analysed. (17 Refs)
Subfile: C
Descriptors: program compilers; storage allocation
Identifiers: register usage penalty; procedure calls; register allocation ; procedure boundaries; one-pass inter-procedural register allocation scheme; depth-first traversal; **call graph** ; priority-based coloring algorithm; usage conventions; recursion; indirect calls; separate compilation; data flow analysis; production compiler
Class Codes: C6150C (Compilers, interpreters and other processors)

23/5/42 (Item 1 from file: 6)
DIALOG(R)File 6:NTIS

2192321 NTIS Accession Number: ADA386599/XAB
Program Flow Graph Construction for Static Analysis of Explicitly Parallel Message-Passing Programs
Shires, D. R. ; Pollock, L.
Courter Products, Boyne City, MI.
Corp. Source Codes: 113876000; 100000
Report No.: ARL-TR-2370
Nov 2000 30p
Languages: English
Journal Announcement: USGRDR0112
Product reproduced from digital image. Order this product from NTIS by: phone at 1-800-553-NTIS (U.S. customers); (703)605-6000 (other countries); fax at (703)605-6900; and email at orders@ntis.gov. NTIS is located at 5285 Port Royal Road, Springfield, VA, 22161, USA.
NTIS Prices: PC A03/MF A01
Country of Publication: United States
In recent years, message-passing parallel codes have rallied around using the message **passing interface** (MPI). The parallelism in these codes is most often explicit; the developer must instrument the source **code** with **calls** to an **optimized** communications runtime library. MPI has been widely used for developing **efficient** and portable parallel **programs** , in particular for distributed memory multiprocessors and workstation/personal computer (PC) clusters, although its use in shared memory systems has been equally **effective** . This report presents algorithm for building a **program flow graph** representation of an MPI program. As an extension of the control flow **graph** representation of sequential codes, this representation provides a basis for important program analyses useful in **software** testing, debugging tools, and **code optimization** .
Descriptors: *Flow charting; *Static tests; Test and evaluation; Computer **programs** ; Algorithms; Control; Sources; **Optimization** ; Distribution; Tools; Sequences; Memory devices; Coding; Construction; Work stations; Multiprocessors; Communication and radio systems; Microcomputers; Time sharing; Debugging(Computers)
Identifiers: Mpi(Message **passing interface**); NTISDODXA
Section Headings: 62B (Computers, Control, and Information Theory--Computer Software)

23/5/48 (Item 1 from file: 144)
DIALOG(R)File 144:Pascal

15387431 PASCAL No.: 02-0075988
Efficient **mapping for message-passing** applications **using the TTIG model: A case study in image processing**
Recent advances in parallel virtual machine and message passing interface **: Santorini/Thera, 23-26 September 2001**
ROIG Concepcio; RIPOLL Ana; BORRAS Javier; LUQUE Emilio
COTRONIS Yiannis, ed; DONGARRA Jack, ed
Universitat de Lleida, Dept. of CS Jaume II 69, 25001 Lleida, Spain; Universitat Autonoma de Barcelona, Dept. of CS, 08193 Bellaterra, Barcelona , Spain
European PVM/MPI Users' Group meeting, 8 (Santorini GRC) 2001-09-23
Journal: Lecture notes in computer science, 2001, 2131 370-377
ISBN: 3-540-42609-4 ISSN: 0302-9743 Availability: INIST-16343; 354000097015340490
No. of Refs.: 8 ref.
Document Type: P (Serial); C (Conference Proceedings) ; A (Analytic)
Country of Publication: Germany
Language: English
In this paper we describe the development and performance of an image

processing application with functional parallelism within the PVM framework. The temporal behaviour of the application is statically modelled with the new task **graph** model TTIG (Temporal Task Interaction **Graph**), that enhances classical models by capturing percentages of concurrency between adjacent tasks. We show how this information can be used in the mapping phase in order to obtain better assignments of tasks to processors. The **effectiveness** of the TTIG in allocation for the **application** under study is established through experimentation on a cluster of PCs.

English Descriptors: Parallelism; Message passing; Parallel programming; Message transmission; Image processing; Performance evaluation; Concurrency; Distributed programming

French Descriptors: Parallelisme; Envoi message; Programmation parallele; Transmission message; Traitement image; Evaluation performance; Simultaneite informatique; Programmation repartie

Classification Codes: 001D02C03

23/5/49 (Item 2 from file: 144)
DIALOG(R)File 144:Pascal

15385955 PASCAL No.: 02-0074482
TOPPER: A tool for optimizing **the performance of parallel** applications **Recent advances in parallel virtual machine and message** passing interface **: Santorini/Thera, 23-26 September 2001**
KONSTANTINOU Dimitris; KOZIRIS Nectarios; PAPAKONSTANTINOU George
COTRONIS Yiannis, ed; DONGARRA Jack, ed
National Technical University of Athens, Dept. of Electrical and Computer Engineering, Computer Science Division Computing Systems Laboratory Zografou Campus, Zografou 15773, Greece
European PVM/MPI Users' Group meeting, 8 (Santorini GRC) 2001-09-23
Journal: Lecture notes in computer science, 2001, 2131 148-157
ISBN: 3-540-42609-4 ISSN: 0302-9743 Availability: INIST-16343; 354000097015340220
No. of Refs.: 11 ref.
Document Type: P (Serial); C (Conference Proceedings) ; A (Analytic)
Country of Publication: Germany
Language: English
In this paper we present an autonomous and complete tool for **optimizing** the performance parallel **programs** on multiprocessor architectures. The concern of TOPPER's users is bound to the construction of two separate **graphs** , describing the overall application's task partitioning and interprocess communication requirements, as well as the architecture of the available multiprocessor system. TOPPER proceeds with the elaboration of these two **graphs** and proposes an **efficient** task mapping, aiming to minimize the **application** 's overall execution time. When the communication between the various tasks is carried out with the use of MPI routines, the tool not only proposes an optimal task allocation but also can execute automatically the parallel application on the target multiprocessing machine.

English Descriptors: Optimal allocation; Execution time; System architecture; Performance evaluation; Multiprocessor; Task scheduling; Parallel program; Software development; Software tool

French Descriptors: Allocation optimale; Temps execution; Architecture systeme; Evaluation performance; Multiprocesseur; Gestion tache; Programme parallele; Developpement logiciel; Outil logiciel; Outil TOPPER

Classification Codes: 001D02B09

23/5/59 (Item 2 from file: 34)
DIALOG(R)File 34:SciSearch(R) Cited Ref Sci

03834470 Genuine Article#: QJ370 Number of References: 14
Title: DISTRIBUTED PROCESS CREATION WITHIN A SHARED DATA SPACE FRAMEWORK
Author(s): ROBINSON PG; ARTHUR JD
Corporate Source: VIRGINIA TECH,DEPT COMP SCI/BLACKSBURG//VA/24060
Journal: SOFTWARE-PRACTICE & EXPERIENCE, 1995, V25, N2 (FEB), P175-191
ISSN: 0038-0644
Language: ENGLISH Document Type: ARTICLE
Geographic Location: USA
Subfile: SciSearch; CC ENGI--Current Contents, Engineering, Technology & Applied Sciences
Journal Subject Category: COMPUTER SCIENCE, SOFTWARE, GRAPHICS, PROGRAMMING
Abstract: This paper describes the design and implementation of a remote process instantiation mechanism which is consistent with the Linda paradigm and semantics of the EVAL operation, and which uses shared data space as the medium for **passing** process and environment **parameters** . The motivation for such an implementation stems from our effort to rehost a uniprocessor version of the Linda computational system to a network of workstations. The baseline version of the Linda system relies on the semantics of the UNIX fork () system **call** to create **processes** and to **pass** the proper execution **parameters** to them. In creating a distributed version of the Linda environment, two major issues are addressed: (1) how to instantiate a remote process that knows where, among several possibilities, to begin its execution, and (2) how to communicate the proper run-time values of relevant variables to each new remote Linda process. Guiding our implementation was a desire to employ existing interprocess communication facilities, i.e. shared data space, to **pass** process creation and execution **parameters** .
Identifiers--KeyWords Plus: SYSTEM; LINDA
Research Fronts: 93-0613 002 (INTERPROCEDURAL DYNAMIC SLICING; PARALLEL MACHINES; NESTED LOOPS; LINDA PROGRAMS; DISTRIBUTED-MEMORY SYSTEMS; SUPERSCALAR COMPILATION; TASK **GRAPHS**)
93-2649 001 (DISTRIBUTED SYSTEMS; LANGUAGE FOR PARALLEL **OPTIMIZATION** ; COMPUTATION MIGRATION)
93-5575 001 (DISTRIBUTED-MEMORY COMPUTERS; LOAD BALANCING DATA-PARALLEL PROGRAMS; EXTENDING UNIX FOR SCALABLE COMPUTING)
Cited References:
C LINDA REFERENCE MA
ANDREWS GR, 1991, V23, P49, COMPUT SURV
BIRRELL AD, 1984, V2, P39, ACM T COMPUT SYST
CARRIERO N, 1989, V32, P444, COMMUN ACM
CLINE G, 1993, P112, 12TH P ANN IEEE INT
GELERNTER D, 1985, V7, P80, ACM T PROGR LANG SYS
GELERNTER D, 1992, V35, P97, COMMUN ACM
HOARE CAR, 1974, V17, P549, COMMUN ACM
HUPFER S, 1991, V574, P187, LECTURE NOTES COMPUT
LANDRY K, 1992, TR9218 VIRG TECH DEP
LELER W, 1992, P43, IEEE COMPUTER FEB
ROBINSON P, 1994, THESIS VIRGINIA TECH
SCHUMANN C, 1991, P49, 1991 P VIRG COMP US
THEIMER MM, 1989, V15, P1444, IEEE T SOFTWARE ENG

23/5/62 (Item 1 from file: 95)
DIALOG(R)File 95:TEME-Technology & Management

00923038 I95080635223
Efficient **implementation of adaptive** software
(Effiziente Implementierung adaptiver Software)
Palsberg, J; Cun Xiao; Lieberherr, K
Coll. of Comput. Sci., Northeastern Univ., Boston, MA, USA
ACM Transactions on Programming Languages and Systems, v17, n2, pp264-292, 1995
Document type: journal article Language: English
Record type: Abstract
ISSN: 0164-0925

ABSTRACT:
Adaptive programs compute with objects, just like object-oriented programs. Each task to be accomplished is specified by a so- **called** propagation pattern which traverses the receiver **object** . The object traversal is a recursive descent via the instance variables where information is collected or propagated along the way. A propagation pattern consists of (1) a name for the task, (2) a succinct specification of the parts of the receiver object that should be traversed, and (3) code fragments to be executed when specific object types are encountered. The propagation patterns need to be complemented by a class **graph** which defines the detailed object structure. The separation of structure and behavior yields a degree of flexibility and understandability not present in traditional object-oriented languages. For example, the class **graph** can be changed without changing the adaptive **program** at all. We present an **efficient** implementation of adaptive **programs** . Given an adaptive **program** and a **class graph** , we generate an **efficient object** -oriented **program** , for example, in C++. Moreover, we prove the correctness of the core of this translation. A key assumption in the theorem is that the traversal specifications are consistent with the class **graph** . We prove the soundness of a proof system for conservatively checking consistency, and we show how to implement it efficiently.

31/5/1 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

11514053 E.I. No: EIP01035582559
Title: Scheduling with bus access optimization for distributed embedded systems
Author: Eles, Petru; Doboli, Alex; Pop, Paul; Peng, Zebo
Corporate Source: Linkoping Univ, Sweden
Conference Title: 11th International Symposium on System-Level Synthesis and Design (ISS'98)
Conference Location: Hsinchu, Taiwan Conference Date: 20981202-20981204
Sponsor: IEEE CS Technical Commitee on Design Automation; ACM SIGDA
E.I. Conference No.: 57950
Source: IEEE Transactions on Very Large Scale Integration (VLSI) Systems v 8 n 5 Oct 2000. p 472-491
Publication Year: 2000
CODEN: IEVSE9 ISSN: 1063-8210
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical); X; (Experimental)
Journal Announcement: 0105W1
Abstract: In this paper, we concentrate on aspects related to the synthesis of distributed embedded systems consisting of programmable processors and application-specific hardware components. The approach is based on an abstract **graph** representation that captures, at process level, both dataflow and the flow of control. Our goal is to derive a worst case delay by which the system completes execution, such that this delay is as small as possible; to generate a logically and temporally deterministic schedule; and to optimize parameters of the communication protocol such that this delay is guaranteed. We have further investigated the impact of particular communication infrastructures and protocols on the overall performance and, specially, how the requirements of such an infrastructure have to be considered for process and communication scheduling. Not only do particularities of the underlying architecture have to be considered during scheduling but also the parameters of the communication protocol should be adapted to fit the particular **embedded application** . The **optimization** algorithm, which implies both process scheduling and optimization of the parameters related to the communication protocol, generates an efficient bus access scheme as well as the schedule tables for activation of processes and communications. (Author abstract) 55 Refs.
Descriptors: *Electric network synthesis; Scheduling; Embedded systems; Distributed computer systems; Microprocessor chips; **Application** specific integrated circuits; **Optimization** ; Network protocols; Algorithms; Real time systems
Identifiers: Distributed embedded systems; Bus access optimization; Abstract **graph** representation; Communication synthesis; System synthesis; Time triggered protocol
Classification Codes:
703.1.2 (Electric Network Synthesis)
703.1 (Electric Networks); 722.4 (Digital Computers & Systems); 714.2 (Semiconductor Devices & Integrated Circuits); 921.5 (Optimization Techniques)
703 (Electric Circuits); 722 (Computer Hardware); 714 (Electronic Components); 921 (Applied Mathematics)
70 (ELECTRICAL ENGINEERING); 72 (COMPUTERS & DATA PROCESSING); 71 (ELECTRONICS & COMMUNICATIONS); 92 (ENGINEERING MATHEMATICS)

31/5/2 (Item 2 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

09984033 E.I. No: EIP04348316106
Title: Automatic design of application specific instruction set

extensions through dataflow graph **exploration**
Author: Clark, Nathan; Zhong, Hongtao; Tang, Wilkin; Mahlke, Scott
Corporate Source: Adv. Comp. Architecture Laboratory University of Michigan, Ann Arbor, MI 48109, United States
Source: International Journal of Parallel Programming v 31 n 6 December 2003. p 429-449
Publication Year: 2003
CODEN: IJPPE5 ISSN: 0885-7458
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 0408W4
Abstract: General-purpose processors are often incapable of achieving the challenging cost, performance, and power demands of high-performance applications. To meet these demands, most systems employ a number of hardware accelerators to off-load the computationally demanding portions of the application. As an alternative to this strategy, we examine customizing the computation capabilities of a processor for a particular application. The processor is extended with hardware in the form of a set of custom function units and instruction set extensions. To effectively identify opportunities for creating custom hardware, a dataflow **graph** design space exploration engine heuristically identifies candidate computation subgraphs without artificially constraining their size or shape. The engine combines estimates of performance gain, cost, and inherent limitations of the processor to grow candidate **graphs** in profitable directions while pruning unprofitable paths. This paper describes the dataflow **graph** exploration engine and evaluates its **effectiveness** across a set of **embedded applications** . 26 Refs.
Descriptors: *Parallel processing systems; Database systems; **Graph** theory; Automatic programming; Cellular telephone systems; Cameras; Program processors; Embedded systems; Personal digital assistants; Mathematical transformations; Algorithms
Identifiers: Application-specific processors; Dataflow **graphs** ; Hardware customization; Instruction sets
Classification Codes:
722.4 (Digital Computers & Systems); 723.3 (Database Systems); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 723.1 (Computer Programming); 716.3 (Radio Systems & Equipment); 742.2 (Photographic Equipment); 921.3 (Mathematical Transformations)
722 (Computer Hardware); 723 (Computer Software, Data Handling & Applications); 921 (Applied Mathematics); 716 (Electronic Equipment, Radar, Radio & Television); 742 (Cameras & Photography)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS); 71 (ELECTRONICS & COMMUNICATION ENGINEERING); 74 (LIGHT & OPTICAL TECHNOLOGY)

31/5/3 (Item 3 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

09492646 E.I. No: EIP03347599513
Title: Interprocedural optimizations for improving data cache performance of array-intensive embedded applications
Author: Zhang, W.; Chen, G.; Kandemir, M.; Karakoy, M.
Corporate Source: CSE Department Pennsylvania State University, University Park, PA 16802, United States
Conference Title: Proceedings of the 40th Design Automation Conference
Conference Location: Anaheim, CA, United States Conference Date: 20030602-20030606
Sponsor: ACM; SIGDA; IEDA Consortium; IEEE; IEEE Circuits and Systems Society; SSCS
E.I. Conference No.: 61299
Source: Proceedings - Design Automation Conference 2003. p 887-892
Publication Year: 2003
CODEN: PDAWDJ ISSN: 0738-100X

Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications); T ; (Theoretical); X; (Experimental)
Journal Announcement: 0308W4
Abstract: As datasets processed by embedded processors increase in size and complexity, the management of higher levels of memory hierarchy (e.g., caches) is becoming an important issue. A major limitation of most of the cache locality optimization techniques proposed by previous research is that they handle a single procedure at a time. This prevents compilers from capturing the data access interactions between procedures and may result in poor performance. In this paper, we look at loop and data transformations from a different angle and use them in an interprocedural optimization framework. Employing the **call graph** representation of a given application, the proposed technique visits each node of this **graph** twice and uses loop and data transformations in a systematic way for **optimizing** array layouts whole **program** wide. Our experimental results show that this interprocedural locality optimization strategy is much more effective than the previous locality-based techniques that handle each procedure in isolation. 11 Refs.
Descriptors: *Embedded systems; Buffer storage; Program processors; Hierarchical systems; Data transfer; Performance; **Graph** theory; Algorithms
Identifiers: Datasets
Classification Codes:
722.1 (Data Storage, Equipment & Techniques); 723.1 (Computer Programming); 731.1 (Control Systems); 723.2 (Data Processing); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory)
722 (Computer Hardware); 723 (Computer Software, Data Handling & Applications); 731 (Automatic Control Principles & Applications); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 73 (CONTROL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

31/5/4 (Item 4 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

09434109 E.I. No: EIP03277531559
Title: Memory- optimized software **synthesis from dataflow** program graphs **with large size data samples**
Author: Oh, Hyunok; Ha, Soonhoi
Corporate Source: Sch. of Elec. Eng./Computer Science Seoul National University, Seoul 151-742, South Korea
Source: Eurasip Journal on Applied Signal Processing v 2003 n 6 May 1 2003. p 514-529
Publication Year: 2003
CODEN: EJASCT ISSN: 1110-8657
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 0307W1
Abstract: In multimedia and graphics applications, data samples of nonprimitive type require significant amount of buffer memory. This paper addresses the problem of minimizing the buffer memory requirement for such applications in **embedded software** synthesis from graphical dataflow programs based on the synchronous dataflow (SDP) model with the given execution order of nodes. We propose a memory minimization technique that separates global memory buffers from local pointer buffers: the global buffers store live data samples and the local buffers store the pointers to the global buffer entries. The proposed algorithm reduces 67% memory for a JPEG encoder, 40% for an H.263 encoder compared with unshared versions, and 22% compared with the previous sharing algorithm for the H.263 encoder. Through extensive buffer sharing **optimization** , we believe that automatic **software** synthesis from dataflow program **graphs** achieves the comparable **code** quality with the manually **optimized code**

in terms of memory requirement. 17 Refs.
Descriptors: *Signal processing; Computer **software** ; Data flow analysis; Synchronization; **Graph** theory; Algorithms; **Optimization**
Identifiers: Buffer memory
Classification Codes:
716.1 (Information & Communication Theory); 723.1 (Computer Programming); 731.1 (Control Systems); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory); 921.5 (Optimization Techniques)
716 (Electronic Equipment, Radar, Radio & Television); 723 (Computer Software, Data Handling & Applications); 731 (Automatic Control Principles & Applications); 921 (Applied Mathematics)
71 (ELECTRONICS & COMMUNICATION ENGINEERING); 72 (COMPUTERS & DATA PROCESSING); 73 (CONTROL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

31/5/5 (Item 5 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

09368393 E.I. No: EIP03187452397
Title: Efficient register and memory assignment for non-orthogonal architectures via graph **coloring and MST algorithms**
Author: Cho, Jeonghun; Paek, Yunheung; Whalley, David
Corporate Source: Electrical Engineering Department Korea Adv. Inst. of Sci./Technology, Daejon 305-701, South Korea
Conference Title: Joint Conference on Languages, Compilers and Tools for Embedded Systems and Software and Compilers for Embedded Systems
Conference Location: Berlin, Germany Conference Date: 20020619-20020621
Sponsor: ACM SIGPLAN
E.I. Conference No.: 60896
Source: Joint COnference on Languages, Compilers and Tools for Embedded Systems and Software and Compilers for Embedded Systems 2002.
Publication Year: 2002
ISBN: 1581135270
Language: English
Document Type: CA; (Conference Article) Treatment: T; (Theoretical); X; (Experimental)
Journal Announcement: 0305W1
Abstract: Finding an optimal assignment of program variables into registers and memory is prohibitively difficult in code generation for application specific instruction-set processors (ASIPs). This is mainly because, in order to meet stringent **speed** and power requirements for **embedded applications** , ASIPs commonly employ non-orthogonal architectures which are typically characterized by irregular data paths, heterogeneous registers and multiple memory banks. As a result, existing techniques mainly developed for relatively regular, orthogonal general-purpose processors (GPPs) are obsolete for these recently emerging ASIP architectures. In this paper, we attempt to tackle this issue by exploiting conventional **graph** coloring and maximum spanning tree (MST) algorithms with special constraints added to handle the non-orthogonality of ASIP architectures. The results in our study indicate that our algorithm finds a fairly good assignment of variables into heterogeneous registers and multi-memories while it runs extremely faster than previous work that employed exceedingly expensive algorithms to address this issue. 15 Refs.
Descriptors: *Program processors; Storage allocation (computer); Computer architecture; Trees (mathematics); Algorithms
Identifiers: General-purpose processors (GPP)
Classification Codes:
723.1 (Computer Programming); 722.1 (Data Storage, Equipment & Techniques); 921.4 (Combinatorial Mathematics, Includes Graph Theory, Set Theory)
723 (Computer Software, Data Handling & Applications); 722 (Computer Hardware); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

31/5/6 (Item 6 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

08923355 E.I. No: EIP01385584550
Title: Synthesis of application-specific coprocessor for core-based ASIC design
Author: Lee, Dae-Hyun; Park, In-Cheol; Kyung, Chong-Min
Corporate Source: Korea Advanced Inst of Science and Technology, South Korea
Source: IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences n 2 2001. p 604-613
Publication Year: 2001
CODEN: IFESEX ISSN: 0916-8508
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 0110W4
Abstract: This paper presents an **efficient** approach for a hardware/**software** partitioning problem: synthesis of an application-specific coprocessor which accelerates an **embedded software** running on a main processor. Given a set of data flow **graphs** (DFGs), most of previous hardware/software partitioning approaches have focused on mapping DFGs to hardware or software. Their common weaknesses are that 1) they ignore various implementation alternatives in realizing DFGs as hardware based on the assumption that only a single hardware implementation exists for a DFG, and that 2) they don't consider the effect of merging on hardware area when synthesizing a coprocessor by merging DFGs. To deal with the first issue, we formulate both the mapping of DFGs to hardware or software and the selection of the appropriate hardware implementation for each DFG as a single integer programming problem, and then apply an iterative algorithm based on the Kernighan and Lin's heuristic to solve the problem. To reduce the CPU time, we have devised data structures that quickly calculate costs of hardware implementations. To deal with the second issue, our method links DFGs with dummy nodes to produce a single large DFG, and then synthesizes a target coprocessor by globally scheduling the DFG and allocating its datapath. Experimental results demonstrate that our approach outperforms the previous approach based on genetic algorithm (GA) in both the coprocessor area and the CPU time. (Author abstract) 12 Refs.
Descriptors: *Program processors; Application specific integrated circuits; Integrated circuit layout; Computer hardware; Data flow analysis; Integer programming; Genetic algorithms; Iterative methods; Data structures
Identifiers: Data flow **graphs** (DFG)
Classification Codes:
723.1 (Computer Programming); 714.2 (Semiconductor Devices & Integrated Circuits); 921.5 (Optimization Techniques); 921.6 (Numerical Methods); 723.2 (Data Processing)
723 (Computer Software, Data Handling & Applications); 714 (Electronic Components & Tubes); 722 (Computer Hardware); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 71 (ELECTRONICS & COMMUNICATION ENGINEERING); 92 (ENGINEERING MATHEMATICS)

31/5/7 (Item 7 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

08645148 E.I. No: EIP00095307913
Title: Partitioning conditional data flow graphs **for embedded system design**
Author: Auguin, M.; Bianco, L.; Capella, L.; Gresset, E.
Corporate Source: Universite de Nice Sophia Antipolis, Sophia-Antipolis, Fr

Conference Title: 2000 IEEE International Conference on Application-Specific Systems, Architectures, and Processors
Conference Location: Boston, MA, USA Conference Date: 19000710-19000712
Sponsor: IEEE Computer Society
E.I. Conference No.: 57187
Source: Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors 2000. IEEE, Piscataway, NJ, USA. p 339-348
Publication Year: 2000
CODEN: PIAAFO
Language: English
Document Type: CA; (Conference Article) Treatment: A; (Applications)
Journal Announcement: 0010W2
Abstract: The complexity of **embedded applications** increases continuously. Integration advances provides a rising range of possibilities to implement a system on a chip. The designers are faced to the difficult challenge to select the right units to implement the application functionalities so that the silicon area is minimized and the time constraints of the **application** are met. This paper presents an **effective method** a design system architectures which operates on a conditional data flow **graph** which is well suited to represent signal processing applications. (Author abstract) 17 Refs.
Descriptors: *Embedded systems; Data flow analysis; Computational complexity; Integrated circuit layout; Semiconducting silicon; Signal processing; Algorithms; Finite automata
Identifiers: Data flow **graph** partitioning; System on a chip
Classification Codes:
712.1.1 (Single Element Semiconducting Materials)
723.5 (Computer Applications); 723.1 (Computer Programming); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory); 714.2 (Semiconductor Devices & Integrated Circuits); 712.1 (Semiconducting Materials); 716.1 (Information & Communication Theory)
723 (Computer Software); 721 (Computer Circuits & Logic Elements); 714 (Electronic Components); 712 (Electronic & Thermionic Materials); 716 (Radar, Radio & TV Electronic Equipment)
72 (COMPUTERS & DATA PROCESSING); 71 (ELECTRONICS & COMMUNICATIONS)

31/5/8 (Item 8 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

08332373 E.I. No: EIP99084745535
Title: Synthesis of embedded software **from synchronous dataflow specifications**
Author: Bhattacharyya, Shuvra S.; Murthy, Praveen K.; Lee, Edward A.
Corporate Source: Univ of Maryland, College Park, MD, USA
Source: Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology v 21 n 2 1999. p 151-166
Publication Year: 1999
CODEN: JVSPED ISSN: 0922-5773
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)
Journal Announcement: 9909W4
Abstract: The implementation of software for embedded digital signal processing (DSP) applications is an extremely complex process. The complexity arises from escalating functionality in the applications; intense time-to-market pressures; and stringent cost, power and speed constraints. To help cope with such complexity, DSP system designers have increasingly been employing high-level, graphical design environments in which system specification is based on hierarchical dataflow **graphs**. Consequently, a significant industry has emerged for the development of data-flow-based DSP design environments. Leading products in this industry

include SPW from Cadence, COSSAP from Synopsys, ADS from Hewlett Packard, and DSP Station from Mentor Graphics. This paper reviews a set of algorithms for compiling dataflow **programs** for embedded DSP **applications** into **efficient** implementations on programmable digital signal processors. The algorithms focus primarily on the minimization of code size, and the minimization of the memory required for the buffers that implement the communication channels in the input dataflow **graph** . These are critical problems because programmable digital signal processors have very limited amounts of on-chip memory, and the speed, power, and cost penalties for using off-chip memory are often prohibitively high for **embedded applications** . Furthermore, memory demands of applications are increasing at a significantly higher rate than the rate of increase in on-chip memory capacity offered by improved integrated circuit technology. (Author abstract) 31 Refs.

Descriptors: *VLSI circuits; Digital signal processing; Software engineering; Specifications; Computational complexity; Algorithms; Data flow analysis; Optimization; Communication channels (information theory); **Graph** theory

Identifiers: Synchronous data flow specifications

Classification Codes:

714.2 (Semiconductor Devices & Integrated Circuits); 716.1 (Information & Communication Theory); 723.1 (Computer Programming); 902.2 (Codes & Standards); 721.1 (Computer Theory, Includes Formal Logic, Automata Theory, Switching Theory, Programming Theory)

714 (Electronic Components); 716 (Radar, Radio & TV Electronic Equipment); 723 (Computer Software); 902 (Engineering Graphics & Standards); 721 (Computer Circuits & Logic Elements); 921 (Applied Mathematics)

71 (ELECTRONICS & COMMUNICATIONS); 72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

31/5/10 (Item 10 from file: 8)

07726944 E.I. No: EIP97063701155

Title: Optimizing synchronization in multiprocessor DSP systems

Author: Bhattacharyya, Shuvra S.; Sriram, Sundararajan; Lee, Edward A.

Corporate Source: Hitachi America, Ltd, San Jose, CA, USA

Source: IEEE Transactions on Signal Processing v 45 n 6 Jun 1997. p 1605-1618

Publication Year: 1997

CODEN: ITPRED ISSN: 1053-587X

Language: English

Document Type: JA; (Journal Article) Treatment: T; (Theoretical)

Journal Announcement: 9708W2

Abstract: This paper is concerned with multiprocessor implementations of **embedded applications** specified as iterative dataflow programs in which synchronization overhead can be significant. We develop techniques to alleviate this overhead by determining a minimal set of processor synchronizations that are essential for correct execution. Our study is based in the context of self-timed execution of iterative dataflow programs. An iterative dataflow program consists of a dataflow representation of the body of a loop that is to be iterated an indefinite number of times; dataflow programming in this form has been studied and applied extensively, particularly in the context of signal processing software. Self-timed execution refers to a combined compile-time/run-time scheduling strategy in which processors synchronize with one another based only on interprocessor communication requirements, and thus, synchronization of processors at the end of each loop iteration does not generally occur. We introduce a new **graph** -theoretic framework based on a data structure called the synchronization **graph** for analyzing and **optimizing** synchronization overhead in self-timed, iterative dataflow **programs** . We show that the comprehensive techniques that have been

developed for removing redundant synchronizations in noniterative programs can be extended in this framework to optimally remove redundant synchronizations in our context. We also present an optimization that converts a feedforward dataflow **graph** into a strongly connected **graph** in such a way as to reduce synchronization overhead without slowing down execution. (Author abstract) 30 Refs.
Descriptors: *Multiprocessing systems; Synchronization; **Optimization** ; Digital signal processing; Iterative **methods** ; Data structures; Computer software
Identifiers: Iterative dataflow program
Classification Codes:
722.4 (Digital Computers & Systems); 731.1 (Control Systems); 921.5 (Optimization Techniques); 716.1 (Information & Communication Theory); 723.2 (Data Processing); 921.6 (Numerical Methods)
722 (Computer Hardware); 731 (Automatic Control Principles); 921 (Applied Mathematics); 716 (Radar, Radio & TV Electronic Equipment); 723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING); 73 (CONTROL ENGINEERING); 92 (ENGINEERING MATHEMATICS); 71 (ELECTRONICS & COMMUNICATIONS)

31/5/13 (Item 13 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

06930550 E.I. No: EIP94091387998
Title: Data routing: A paradigm for efficient **data-path synthesis and** code **generation**
Author: Lanneer, Dirk; Cornero, Marco; Goossens, Gert; De Man, Hugo
Corporate Source: IMEC, Leuven, Belgium
Conference Title: Proceedings of the 7th International Symposium on High-Level Synthesis
Conference Location: Niagara-on-the-Lake, Ont, Can Conference Date: 19940518-19940520
Sponsor: IEEE Computer Society; SIGDA
E.I. Conference No.: 20763
Source: Proceedings of the IEEE International Symposium on High-Level Synthesis 1994. Publ by IEEE, Computer Society Press, Los Alamitos, CA, USA,94TH0641-1. p 17-22
Publication Year: 1994
CODEN: 001582 ISBN: 0-8186-5785-5
Language: English
Document Type: CA; (Conference Article) Treatment: G; (General Review); T; (Theoretical)
Journal Announcement: 9410W2
Abstract: This paper describes a new and effective approach to register and interconnect optimization, which is applicable in a dual context: to reduce chip area in high-level synthesis, and to reduce resource load (and thus execution time) in retargetable **code** generation. The key idea is to carefully **optimize** the way in which data is transferred between functional units. The impact on high-level synthesis will be demonstrated with a practical design from the area of telecommunications. (Author abstract) 17 Refs.
Descriptors: *Critical path analysis; Codes (symbols); Microprocessor chips; Storage allocation (computer); VLSI circuits; Mathematical transformations; **Program** processors; **Optimization** ; Data transfer
Identifiers: Data routing; Data path synthesis; Retargetable code generation; High level synthesis; Register allocation; **Embedded application** specific digital signal processors; Control data flow **graph**
Classification Codes:
723.2 (Data Processing); 714.2 (Semiconductor Devices & Integrated Circuits); 721.3 (Computer Circuits); 921.3 (Mathematical Transformations); 723.1 (Computer Programming); 921.5 (Optimization Techniques)
723 (Computer Software); 714 (Electronic Components); 721 (Computer

technology. We propose a register-transfer level leakage power reduction technique based on leakage estimation and dual-threshold design libraries.

We have seen that today's **embedded applications** typically work with large data sets which are stored in memories. The memory accesses performed by an application are represented as array accesses in the behavioral specification. The manner in which arrays in the behavior are mapped to physical memory (memory binding) is an important determinant of the performance of the design. In other words, the quality of the generated schedule depends upon the binding information assumed by the scheduler. We introduce a memory binding algorithm that preserves the parallelism in the application so that a high-performance schedule can be derived for it. Our technique accepts as inputs, a control-flow intensive behavior, memory allocation information, and functional unit allocation information, and returns a mapping of arrays in the functional specification to the allocated memories. The mapping is performed with the objective of optimizing the performance of the scheduled behavior. However, performance is also highly affected by the clock period chosen for the design. We also present a technique that performs clock selection for optimizing the performance of control-flow intensive behaviors.

In summary, we present a suite of techniques that target high-level synthesis of performance- and power-optimized control-flow intensive architectures.

31/5/17 (Item 2 from file: 35)

01209291 ORDER NO: AAD92-08398
AN APPROACH TO PERFORMANCE DEPENDENCY **ANALYSIS FOR DISTRIBUTED** EMBEDDED SOFTWARE **SYSTEMS (ADA)**
Author: HSIEH, CHIN-YUN
Degree: PH.D.
Year: 1991
Corporate Source/Institution: THE UNIVERSITY OF OKLAHOMA (0169)
Major Professor: PAO-SHENG E. CHANG
Source: VOLUME 52/10-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 5367. 135 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

Retaining a satisfactory performance for the maintenance of a distributed **embedded software** system has been an important, yet difficult issue. This issue can be largely attributed to a so called performance ripple effect phenomenon of program modifications. The propagation of this phenomenon, however, is essentially caused by the performance dependencies between program entities.

In this dissertation, an approach to the analysis of performance dependencies for distributed **embedded software** systems is presented. Ada programming language is chosen as the target language to illustrate this apporach. The entire analysis process comprizes two phases: the analyzing phase and the tracing phase. In the analyzing phase, an Ada source program will be analyzed and those linguistic mechanisms whose uses cause performance dependencies between program entities will be identified. An extended module performance **dependency** (EMPD) **graph** model will then be constructed. In this model, each syntactically independent module will be partitioned into primitive program entities which will then be mapped into vertices of this **graph** model. Edges between vertices represent direct performance **dependency** relations between corresponding program entities. One important property of this EMPD model is that it preserves the transitive property of performance **dependency** relation between program entities. Tracing the indirect performance dependencies can then be converted into a simple computation of transitive closure of this EMPD model.

Furthermore, as Ada is a structured programming language, the semantic

relations between modules are constrained to the scoping of the involved modules. Accordingly, a heuristic algorithm is presented for the analysis of indirect performance dependencies between program entities. In this algorithm, performance dependencies between entities within each module will first be analyzed. Global performance **dependency** information will then be extracted and used when the module participates in the upper level performance **dependency** analysis. This tracing process starts from the innermost nested executable program entity, proceeds recursively, and terminates at the outermost level **modules** . This approach is much more **efficient** than an extensive computation of the transitive closure of the entire EMPD. This technique forms partially the theoretical basis for the analysis of performance ripple effect as a consequence of program modifications.

31/5/20 (Item 1 from file: 2)
DIALOG(R)File 2:INSPEC

09131871 INSPEC Abstract Number: B2004-11-1265A-037, C2004-11-5215-011
Title: High-level synthesis based upon dependence graph **for multi-FPGA**
Author(s): Akil, M.
Author Affiliation: Lab. A2SI, Groupe ESIEE, Noisy Le Grand, France
Journal: Informacije MIDEM vol.33, no.4 p.267-75
Publisher: Soc. Microelectron. Electron. Components & Mater.-MIDEM,
Publication Date: 2003 Country of Publication: Slovenia
CODEN: IMIDEN ISSN: 0352-9045
SICI: 0352-9045(2003)33:4L.267:HLSB;1-V
Material Identity Number: N527-2004-003
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: The increasing complexity of signal, image and control processing algorithms in real-time **embedded applications** requires **efficient** system-level design methodology to help the designer to solve the specification, validation and synthesis problems. Indeed, the real-time and embedded constraints may be so strong that the available high performance processors are not sufficient. That leads to use, in complement of processor, the specific component like ASIC or FPGA. Several projects have developed high-level design flow that translates high-level algorithm specification to an efficient implementation for mapping onto multi-component architecture. In this paper, we present: 1. a unified model for hardware/software codesign, based on the AAA methodology (algorithm-architecture adequation). In order to exhibit the potential parallelism of algorithm to be implemented, the AAA methodology is based on conditioned (conditional execution of computations) factorized (loop) data dependence **graph** . 2. Some simple rules that allow synthesizing both the data path and the control path of a circuit corresponding to an algorithm specified as a conditioned and factorized data dependence **graph** (CFDDG). 3. The optimized implementation of CFDDG algorithm onto FPGA circuit and multi FPGA (partitioning), by using simulated annealing approach. 4. The resources and time delay estimation method. This method allows us to have a performance analysis for the implementation. The obtained results (resource estimation, latency estimation) are used by the optimization step to decide which implementation respects the constraints (real-time implementation which minimises the resource utilisation). 5. The results of the implementation of the matrix-vector product algorithm onto a Xilinx multi FPGA and the software tool SynDEx which implements the AAA methodology. (12 Refs)
Subfile: B C
Descriptors: embedded systems; field programmable gate arrays; hardware description languages; hardware-software codesign; simulated annealing
Identifiers: high-level synthesis; signal processing algorithm; image processing algorithm; control processing algorithm; real-time **embedded applications** ; system-level design; formal specification; formal validation ; ASIC; high-level design flow; multicomponent architecture; unified model;

hardware software codesign; AAA methodology; algorithm-architecture adequation; computation conditional execution; circuit data path; circuit control path; conditioned and factorized data dependence **graph** ; FPGA circuit; simulated annealing; resource estimation; time delay estimation; performance analysis; latency estimation; real-time implementation; resource utilisation; matrix-vector product algorithm; Xilinx multiFPGA; software tool SynDEx

Class Codes: B1265A (Digital circuit design, modelling and testing); B1265B (Logic circuits); B0260 (Optimisation techniques); C5215 (Hardware-software codesign); C7410D (Electronic engineering computing); C5120 (Logic and switching circuits); C1180 (Optimisation techniques)

31/5/21 (Item 2 from file: 2)

DIALOG(R)File 2:INSPEC

09015788 INSPEC Abstract Number: C2004-08-6110B-059

Title: GA based inlining optimization in front-end synthesis of embedded software

Author(s): Min Li; Hui Wang; Xiaohong Zhu; Ping Li

Author Affiliation: Sch. of Inf. Sci. & Eng., Zhejiang Univ., Hangzhou, China

Conference Title: 2003 5th International Conference on ASIC. Proceedings (IEEE Cat. No.03TH8690) Part Vol.1 p.341-3 Vol.1

Editor(s): Tang, T.A.; Wenhong, L.; Yu, H.

Publisher: IEEE, Piscataway, NJ, USA

Publication Date: 2003 Country of Publication: USA 1478 pp.

ISBN: 0 7803 7889 X Material Identity Number: XX-2004-00196

U.S. Copyright Clearance Center Code: 0-7803-7889-X/03/$17.00

Conference Title: 2003 5th International Conference on ASIC. Proceedings

Conference Sponsor: Chinese Inst of Electron.; IEEE Beijing Section; IEE Beijing Branch

Conference Date: 21-24 Oct. 2003 Conference Location: Beijing, China

Language: English Document Type: Conference Paper (PA)

Treatment: Applications (A); Theoretical (T); Experimental (X)

Abstract: The front-end **optimization** of **embedded software** has attracted a lot of interest in recent years. Since most computation is consumed in small fraction of code called hot path or computation kernel, even a bit performance improvement in hot path can result in great promotion, hence, function inlining become one of the effective techniques because taking away the function calling overhead will speed up the hot path. Most of previous work inlines functions globally, and also causes unwanted code increment in cold path. In our approach to function inlining, Accurate Functions **Dependency Graph** (AFDG) is presented to accurately modeling the function calling behavior, and the inlining optimization is carried out according to profiling data on AFDG, so that functions are partially inlined, and code increment in cold path is prevented. In order to get a satisfactory solution from the tremendous solution space in short time, a meta-heuristic genetic algorithm (GA) is applied. In experiments, the proposed algorithm shows promising results compared with previous work. (13 Refs)

Subfile: C

Descriptors: embedded systems; genetic algorithms; set theory

Identifiers: GA based inlining optimization; genetic algorithm; front end synthesis; **embedded software** ; front end optimization; hot path; computation kernel; function calling; cold path; function inlining; accurate functions **dependency graph** ; AFDG; code increment; meta heuristic genetic algorithm; set theory

Class Codes: C6110B (Software engineering techniques); C1180 (Optimisation techniques); C1160 (Combinatorial mathematics)

31/5/22 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC

08962751 INSPEC Abstract Number: B2004-06-1265F-078, C2004-06-5130-061
Title: Processor acceleration through automated instruction set customization
Author(s): Clark, N.; Hongtao Zhong; Mahlke, S.
Author Affiliation: Adv. Comput. Archit. Lab., Michigan Univ., Ann Arbor, MI, USA
Conference Title: 36th International Symposium on Microarchitecture p. 129-40
Publisher: IEEE Comput. Soc, Los Alamitos, CA, USA
Publication Date: 2003 Country of Publication: USA xiv+436 pp.
ISBN: 0 7695 2043 X Material Identity Number: XX-2004-00102
U.S. Copyright Clearance Center Code: 0-7695-2043-X/03/$17.00
Conference Title: 36th International Symposium on Microarchitecture
Conference Sponsor: IEEE TC-MARCH; SIGMICRO
Conference Date: 3-5 Dec. 2003 Conference Location: San Diego, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Application-specific extensions to the computational capabilities of a processor provide an efficient mechanism to meet the growing performance and power demands of **embedded applications** . Hardware, in the form of new function units (or co-processors), and the corresponding instructions, are added to a baseline processor to meet the critical computational demands of a target application. The central challenge with this approach is the large degree of human effort required to identify and create the custom hardware units, as well as porting the application to the extended processor. In this paper, we present the design of a system to automate the instruction set customization process. A dataflow **graph** design space exploration engine efficiently identifies profitable computation subgraphs from which to create custom hardware, without artificially constraining their size or shape. The system also contains a compiler subgraph matching framework that identifies opportunities to exploit and generalize the hardware to support more computation **graphs** . We demonstrate the **effectiveness** of this system across a range of **application** domains and study the applicability of the custom hardware across the domain. (33 Refs)
Subfile: B C
Descriptors: application specific integrated circuits; data flow **graphs** ; instruction sets; logic design; program compilers
Identifiers: processor acceleration; automated instruction set customization; application-specific extensions; processor computational capabilities; **embedded applications** ; function units; baseline processor ; hardware units; dataflow **graph** design space exploration engine; computation subgraphs; compiler subgraph matching framework; computation **graphs**
Class Codes: B1265F (Microprocessors and microcomputers); B1265A (Digital circuit design, modelling and testing); B0250 (Combinatorial mathematics); C5130 (Microprocessor chips); C5210 (Logic design methods); C1160 (Combinatorial mathematics); C6150C (Compilers, interpreters and other processors)

31/5/23 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC

08857657 INSPEC Abstract Number: B2004-03-6135C-097, C2004-03-6120-019
Title: Memory- optimized software **synthesis from dataflow** program graphs **with large size data samples**
Author(s): Hyunok Oh; Soonhoi Ha
Author Affiliation: Sch. of Electr. Eng. & Comput. Sci., Seoul Nat.

Univ., South Korea
Journal: EURASIP Journal on Applied Signal Processing vol.2003, no.6 p.514-29
Publisher: Hindawi,
Publication Date: 1 May 2003 Country of Publication: USA
CODEN: EJASCT ISSN: 1110-8657
SICI: 1110-8657(20030501)2003:6L.514:MOSS;1-S
Material Identity Number: H080-2003-008
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P); Experimental (X)
Abstract: In multimedia and graphics applications, data samples of nonprimitive type require significant amount of buffer memory. This paper addresses the problem of minimizing the buffer memory requirement for such applications in **embedded software** synthesis from graphical dataflow programs based on the synchronous dataflow (SDF) model with the given execution order of nodes. We propose a memory minimization technique that separates global memory buffers from local pointer buffers: the global buffers store live data samples and the local buffers store the pointers to the global buffer entries. The proposed algorithm reduces 67% memory for a JPEG encoder, 40% for an H.263 encoder compared with unshared versions, and 22% compared with the previous sharing algorithm for the H.263 encoder. Through extensive buffer sharing **optimization** , we believe that automatic **software** synthesis from dataflow program **graphs** achieves the comparable **code** quality with the manually **optimized code** in terms of memory requirement. (17 Refs)
Subfile: B C
Descriptors: buffer storage; computer graphics; data flow **graphs** ; embedded systems; image coding; minimisation; multimedia computing
Identifiers: memory optimization; multimedia applications; graphics applications; **embedded software** synthesis; synchronous dataflow model; memory minimization technique; global memory buffers; local pointer buffers ; JPEG; joint photographic experts group encoder; buffer sharing optimization; automatic software
Class Codes: B6135C (Image and video coding); B0250 (Combinatorial mathematics); B0260 (Optimisation techniques); C6120 (File organisation); C5260B (Computer vision and image processing techniques); C1160 (Combinatorial mathematics); C6130B (Graphics techniques); C6130M (Multimedia); C1180 (Optimisation techniques)

31/5/28 (Item 9 from file: 2)
DIALOG(R)File 2:INSPEC

08307933 INSPEC Abstract Number: C2002-08-6110V-008
Title: Call graph **and control flow** graph **visualization for developers of** embedded applications
Author(s): Evstiougov-Babaev, A.A.
Author Affiliation: AbsInt Angewandte Informatik GmbH, Saarbrucken, Germany
Conference Title: Software Visualization. International Seminar. Revised Papers (Lecture Notes in Computer Science Vol.2269) p.337-46
Editor(s): Diehl, S.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 2002 Country of Publication: Germany viii+403 pp.
ISBN: 3 540 43323 6 Material Identity Number: XX-2002-01069
Conference Title: Software Visualization. International Seminar. Revised Papers
Conference Date: 20-25 May 2001 Conference Location: Dagstuhl Castle, Germany
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: When working with complex **software** , visualization improves understanding considerably. Thus, **enhancing** the ability of programmers to

picture the relationships between components in a complex program not only saves time but becomes progressively mission-critical with increasing software complexity. aiCall is a software visualization tool which helps programmers to better understand their software, generally improving learning, speeding up development and saving considerable effort and expense. aiCall visualizes the **call graph** and the control flow **graph** of **embedded application** code. Currently supported targets are Infineon C16* (Evstiougov-Babaev and Ferdinand, 2001) and STMicroelectronics ST10. These microcontroller families are very popular and widely used in consumer goods (cellular phones, CD-players, washing machines) and in safety-critical environments (airbags, navigation systems, and automotive controls). (15 Refs)
Subfile: C
Descriptors: embedded systems; flow **graphs** ; program visualisation; reverse engineering; software tools
Identifiers: **call graph** visualization; control flow **graph** visualization; **embedded applications** ; complex software; program understanding; Infineon C16; STMicroelectronics ST10; microcontroller; safety-critical environments; software components; software complexity; aiCall; software visualization tool
Class Codes: C6110V (Visual programming); C6115 (Programming support); C1160 (Combinatorial mathematics)

31/5/29 (Item 10 from file: 2)
DIALOG(R)File 2:INSPEC

08306226 INSPEC Abstract Number: C2002-08-5220-002
Title: Consistency analysis of reconfigurable dataflow specifications
Author(s): Bhattacharya, B.; Bhattacharyya, S.S.
Author Affiliation: Cadence Design Syst. Inc., San Jose, CA, USA
Book Title: Embedded processor design challenges. Systems, architectures, modeling, and simulation - SAMOS (Lecture Notes in Computer Science Vol.2268) p.1-17
Editor(s): Deprettere, E.F.; Teich, J.; Vassiliadis, S.
Publisher: Springer-Verlag, Berlin, Germany
Publication Date: 2002 Country of Publication: Germany viii+325 pp.
ISBN: 3 540 43322 8 Material Identity Number: XB-2002-00027
Language: English Document Type: Book Chapter (BC)
Treatment: Theoretical (T)
Abstract: Parameterized dataflow is a meta-modeling approach for incorporating dynamic reconfiguration capabilities into broad classes of dataflow-based design frameworks for digital signal processing (DSP). Through a novel formalization of dataflow parameterization, and a disciplined approach to specifying parameter reconfiguration, the parameterized dataflow framework provides for automated synthesis of robust and **efficient embedded software** . Central to these synthesis objectives is the formulation and analysis of consistency in parameterized dataflow specifications. Consistency analysis of reconfigurable specifications is particularly challenging due to their inherently dynamic behavior. This paper presents a novel framework, based on a concept of local synchrony, for managing consistency when synthesizing implementations from dynamically-reconfigurable, parameterized dataflow **graphs** . (18 Refs)
Subfile: C
Descriptors: data flow computing; data flow **graphs** ; digital signal processing chips; reconfigurable architectures
Identifiers: parameterized dataflow; dynamic reconfiguration; digital signal processing; dataflow-based design frameworks; consistency analysis; reconfigurable specifications; local synchrony
Class Codes: C5220 (Computer architecture); C5260 (Digital signal processing); C5135 (Digital signal processing chips); C5210 (Logic design methods)

31/5/30 (Item 11 from file: 2)

07851522 INSPEC Abstract Number: B2001-04-1265F-005, C2001-04-5130-002

Title: Synthesis of application-specific coprocessor for core-based ASIC design

Author(s): Dae-Hyun Lee; In-Cheol Park; Chong-Min Kyung
Author Affiliation: Korea Adv. Inst. of Sci. & Technol., Seoul, South Korea

Journal: IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences vol.E84-A, no.2 p.604-13
Publisher: Inst. Electron. Inf. & Commun. Eng,
Publication Date: Feb. 2001 Country of Publication: Japan
CODEN: IFESEX ISSN: 0916-8508
SICI: 0916-8508(200102)E84A:2L.604:SASC;1-8
Material Identity Number: P710-2001-003
Language: English Document Type: Journal Paper (JP)
Treatment: Applications (A); Practical (P)

Abstract: This paper presents an **efficient** approach for a hardware/**software** partitioning problem: synthesis of an application-specific coprocessor which accelerates an **embedded software** running on a main processor. Given a set of data flow **graphs** (DFGs), most previous hardware/software partitioning approaches have focused on mapping DFGs to hardware or software. Their common weaknesses are that (1) they ignore various implementation alternatives in realizing DFGs as hardware based on the assumption that only a single hardware implementation exists for a DFG, and that (2) they don't consider the effect of merging on hardware area when synthesizing a coprocessor by merging DFGs. To deal with the first issue, we formulate both the mapping of DFGs to hardware or software and the selection of the appropriate hardware implementation for each DFG as a single integer programming problem, and then apply an iterative algorithm based on Kernighan and Lin's heuristic to solve the problem. To reduce the CPU time, we have devised data structures that quickly calculate costs of hardware implementations. To deal with the second issue, our method links DFGs with dummy nodes to produce a single large DFG, and then synthesizes a target coprocessor by globally scheduling the DFG and allocating its datapath. Experimental results demonstrate that our approach outperforms the previous approach based on genetic algorithm (GA) in both the coprocessor area and the CPU time. (12 Refs)

Subfile: B C
Descriptors: application specific integrated circuits; coprocessors; data flow **graphs** ; genetic algorithms; hardware-software codesign; integer programming; integrated circuit design; iterative methods
Identifiers: application-specific coprocessor; core-based ASIC; ASIC design; hardware/software partitioning problem; **embedded software** ; data flow **graphs** ; merging; integer programming problem; iterative algorithm; Kernighan and Lin's heuristic; data structures; dummy nodes; target coprocessor; genetic algorithm
Class Codes: B1265F (Microprocessors and microcomputers); B0290F (Interpolation and function approximation (numerical analysis)); B0250 (Combinatorial mathematics); B2570A (Semiconductor integrated circuit design, layout, modelling and testing); B1265A (Digital circuit design, modelling and testing); B0260 (Optimisation techniques); C5130 (Microprocessor chips); C4130 (Interpolation and function approximation (numerical analysis)); C1160 (Combinatorial mathematics); C7410D (Electronic engineering computing); C5215 (Hardware-software codesign); C1180 (Optimisation techniques)

31/5/32 (Item 13 from file: 2)

DIALOG(R)File 2:INSPEC

07332726 INSPEC Abstract Number: C1999-10-6150C-001
Title: Code Generation for Embedded Processors
Journal: Design Automation for Embedded Systems vol.4, no.2-3
Publisher: Kluwer Academic Publishers,
Publication Date: March 1999 Country of Publication: Netherlands
CODEN: DAESFC ISSN: 0929-5585
Material Identity Number: E388-1999-003
U.S. Copyright Clearance Center Code: 99/$9.50
Conference Title: Code Generation for Embedded Processors
Conference Date: 4-6 March 1998 Conference Location: Witten, Germany
Language: English Document Type: Conference Proceedings (CP); Journal Paper (JP)
Abstract: This workshop focused on techniques for designing compilers for embedded systems. This topic is becoming extremely important due to the trends in embedded system design. More and more embedded systems are implemented with major amounts of software. The main advantage of software is flexibility. Changes of design specifications can be taken into account by recompiling changed specifications. For ASICs, design changes are much more costly. Specifications, especially in the DSP world, are frequently written in C. This means that compilers are required for translating into machine code. Unfortunately, standard compiler technology does not support many of the architectures of embedded systems very well. Embedded systems frequently have to be **efficient** (for example, in terms of power or **code** density). Hence, they include features (like specialised registers) which make them more efficient, but less regular. Traditional compiler concepts do not support irregular architectures well. Furthermore, they ignore special concepts of application domains such as DSP. Hence, they generate inefficient code. This cannot be tolerated, especially for portable equipment and hence, assembly language programming is still wide-spread. Related problems exist for compilers for very long instructions word (VLIW) machines. For VLIW machines and for irregular architectures, there is a strong **dependency** between the different compiler phases and new techniques for representing the solution space as well as for coupling compiler phases are required. In addition, new optimization techniques are required in order to make compiled code competitive for embedded processors. Many of the authors at the workshop aim at replacing assembly-language programming for embedded systems. Software generation for embedded systems also involves many other issues: the mapping of applications to multi-processor systems; validation of embedded systems comprising **embedded software** ; analysis of timing issues; and reliability of **embedded software** . These and similar issues were discussed at the series of workshops, for which this special issue presents some of the results.
Subfile: C
Descriptors: embedded systems; program compilers
Identifiers: compilers; embedded systems; design specifications; ASICs; DSP; compiler technology; mapping; timing issues; validation; reliability; code generation
Class Codes: C6150C (Compilers, interpreters and other processors)

31/5/33 (Item 14 from file: 2)
DIALOG(R)File 2:INSPEC

06969013 INSPEC Abstract Number: C9808-5210B-062
Title: A unified model for software-hardware co-design
Author(s): Lavarenne, C.; Sorel, Y.
Author Affiliation: Inst. Nat. de Recherche en Inf. et Autom., Le Chesnay, France
Journal: Traitement du Signal vol.14, no.6 p.569-78

Publisher: GRETSI,
Publication Date: 1997 Country of Publication: France
CODEN: TRSIE6 ISSN: 0765-0019
SICI: 0765-0019(1997)14:6L.569:UMSH;1-A
Material Identity Number: H686-98004
Language: French Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: A unified model of factorized **graphs** is proposed for the specification and the **optimization** of real-time **embedded applications** based on architectures composed of processors and/or specific circuits. First, a **graph** of operations partially ordered by their data dependencies is used to specify the algorithm and hence its potential parallelism, independently of hardware constraints. Then, it is shown how this dependence **graph** may be transformed by different kinds of factorization to obtain an implementation, as specific circuits or as a specialized executive distributed on several processors. Finally, basic principles of optimization are given for minimizing hardware resources while satisfying real-time constraints. In prospect, this unified approach is expected to be used for **optimized software** -hardware co-design. (11 Refs)
Subfile: C
Descriptors: computer aided software engineering; formal specification; **graph** theory; high level synthesis; optimisation; real-time systems
Identifiers: unified model; software hardware codesign; factorized **graphs** ; specification; optimization; real-time **embedded applications** ; data dependencies; parallel algorithm; dependence **graph**0
Class Codes: C5210B (Computer-aided logic design); C7410D (Electronic engineering computing); C6110F (Formal methods); C6115 (Programming support)

31/5/34 (Item 15 from file: 2)
DIALOG(R)File 2:INSPEC

06309466 INSPEC Abstract Number: C9608-6110B-007
Title: Operation serializability for embedded systems
Author(s): Gupta, R.K.
Author Affiliation: Dept. of Comput. Sci., Illinois Univ., Urbana, IL, USA
Conference Title: Proceedings. European Design and Test Conference ED&TC 96 (Cat. No.96TB100027) p.108-14
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1996 Country of Publication: USA xxxi+623 pp.
ISBN: 0 8186 7423 7 Material Identity Number: XX95-03166
U.S. Copyright Clearance Center Code: 1066-1409/96/$5.00
Conference Title: Proceedings of European Design and Test Conference
Conference Sponsor: IEEE Comput. Soc.; Eur. Design & Autom. Assoc.; Eur. Group of TTTC & DATC; ACM/SIGDA; Eur. Comission; ADFTT, NIS (Ex Soviet Union); AEIA, Spain; AFCET, France; ATI, Spain; BULL, France; CLRC, UK; CNR, Italy; CSIC, Spain; Estonian Electron. Soc., Estonia; GI, Germany; HTE, Hungary; IEEE Circuits & Syst. Soc.; IFIP, 10.2/10.5; ITG, Germany; KVIV, Belgium; MATE, Hungary; Polish Acad. Sci., Poland; VDE, Germany
Conference Date: 11-14 March 1996 Conference Location: Paris, France
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Theoretical (T)
Abstract: We consider the problem of generation of **embedded software** from input system descriptions in a hardware description language (HDL). Generation of software for embedded computing requires a total ordering of operations, or linearization, under constraints to ensure timely interaction with other system components. We show by example conditions where no ordering of operations in a HDL can produce the modeled functionality in software. Therefore, the existence condition for software generation or serializability must be ensured before attempting any linearization. We present the conditions based on variable definition and

use analysis under which operation linearization is possible. We then present our approach to operation serialization under timing constraints to produce **efficient** schedules for the **embedded software** . (15 Refs)
Subfile: C
Descriptors: flow **graphs** ; hardware description languages; linearisation techniques; real-time systems; software engineering
Identifiers: operation serializability; embedded systems; hardware description language; total ordering; linearization; software functionality ; software generation; timing constraints
Class Codes: C6110B (Software engineering techniques); C1160 (Combinatorial mathematics); C6140D (High level languages)

31/5/35 (Item 16 from file: 2)
DIALOG(R)File 2:INSPEC

06224584 INSPEC Abstract Number: C9605-5220-011
Title: An evolution-based technique for local microcode compaction
Author(s): Ahmad, I.; Dhodhi, M.K.; Saleh, K.A.
Author Affiliation: Dept. of Electr. & Comput. Eng., Kuwait Univ., Safat, Kuwait
Conference Title: Proceedings of the ASP-DAC`95/CHDL`95/VLSI`95. Asia and South Pacific Design Automation Conference. IFIP International Conference on Computer Hardware Description Languages and their Applications. IFIP Interntional Conference on Very Large Scale Integration (IEEE Cat. No.95TH8102) p.729-34
Publisher: Nihon Gakkai Jimu Senta, Tokyo, Japan
Publication Date: 1995 Country of Publication: Japan xxxii+860 pp.
ISBN: 4 930813 67 0 Material Identity Number: XX94-02583
Conference Title: Proceedings of ASP-DAC`95/CHDL`95/VLSI`95 with EDA Technofair
Conference Sponsor: IFIP WG 10.5 (Former 10.2 & 10.5); IEICE; IPSJ (Inf. Process. Soc. Japan); ACM SIGDA; IEEE Circuits & Syst. Soc.; IEEE Comput. Soc
Conference Date: 29 Aug.-1 Sept. 1995 Conference Location: Chiba, Japan
Language: English Document Type: Conference Paper (PA)
Treatment: Theoretical (T)
Abstract: In this paper we present a variant of the simulated evolution technique for local microcode compaction. The simulated evolution is a general **optimization method** based on an analogy with the natural selection process in biological evolution. The proposed technique combines simulated evolution with list scheduling, in which simulated evolution is used to determine suitable priorities which lead to a good solution by applying list scheduling as a decoding heuristic. The proposed technique is an **effective method** that yields good results without problem-specific parameter tuning on test problems. We demonstrate the effectiveness of our technique by comparing it with the existing microcode compaction techniques for randomly generated data **dependency graphs** . The proposed scheme offers considerable improvement in the number of microinstructions compared with the existing techniques with comparable cpu time. (18 Refs)
Subfile: C
Descriptors: **firmware** ; genetic algorithms; microprogramming; scheduling
Identifiers: microcode compaction; **optimization method** ; simulated evolution; list scheduling; decoding heuristic; microinstructions
Class Codes: C5220 (Computer architecture); C1180 (Optimisation techniques); C5140 (Firmware)

31/5/36 (Item 17 from file: 2)
DIALOG(R)File 2:INSPEC

06159508 INSPEC Abstract Number: C9602-5220-005
Title: An evolutionary technique for local microcode compaction
Author(s): Ahmad, I.; Dhodhi, M.K.; Saleh, K.A.
Author Affiliation: Dept. of Electr. & Comput. Eng., Kuwait Univ., Safat, Kuwait
Journal: Microprocessors and Microsystems vol.19, no.8 p.467-74
Publisher: Elsevier,
Publication Date: Oct. 1995 Country of Publication: UK
CODEN: MIMID5 ISSN: 0141-9331
SICI: 0141-9331(199510)19:8L.467:ETLM;1-4
Material Identity Number: M242-96001
U.S. Copyright Clearance Center Code: 0141-9331/95/$9.50
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: In this paper we present a variant of the simulated evolution technique for local microcode compaction. Simulated evolution is a general **optimization method** based on an analogy with the natural selection process in biological evolution. The proposed technique combines simulated evolution with list scheduling, in which simulated evolution is used to determine suitable priorities which lead to a good solution by applying list scheduling as a decoding heuristic. The proposed technique is an **effective method** that yields good results without problem-specific parameter tuning on test problems of very different sizes and structures. This is achieved by establishing a reasonable balance between exploration of the search space and exploitation of good solutions found in an acceptable CPU time. We demonstrate the effectiveness of our technique by comparing it with the existing microcode compaction techniques for randomly generated data **dependency graphs** . The proposed scheme offers considerable improvement in the number of microinstructions compared with the existing techniques with comparable CPU time. (16 Refs)
Subfile: C
Descriptors: **firmware** ; optimisation
Identifiers: microcode compaction; simulated evolution; optimization; list scheduling; search space
Class Codes: C5220 (Computer architecture); C5140 (Firmware); C1180 (Optimisation techniques)

31/5/37 (Item 18 from file: 2)
DIALOG(R)File 2:INSPEC

04248551 INSPEC Abstract Number: C88064360
Title: Towards portable microcode
Author(s): Boring, R.E.; Andrews, M.; Lam, F.
Author Affiliation: Space Tech Corp., Fort Collins, CO, USA
Conference Title: IEEE Region 5 Conference 1988: Spanning the Peaks of Electrotechnology (Cat. No.88CH2567-6) p.1-5
Publisher: IEEE, New York, NY, USA
Publication Date: 1988 Country of Publication: USA xii+248 pp.
U.S. Copyright Clearance Center Code: CH2567-6/88/0000-0001$01.00
Conference Sponsor: IEEE
Conference Date: 21-23 March 1988 Conference Location: Colorado Springs, CO, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: The implementation of a retargetable microcode generator system (RMGS) is described for a six-stage translation system. Portability of microcode is achieved by specifying microprograms in a C-like HLL (high-level language) and translating the specification into horizontal microcode. The HLL microprogram description is translated into an intermediate semantic description language (ISDL). The ISDL specification is then translated into a machine-dependent microcode using a heuristic

pattern-matched code generator. Retargetability to a wide variety of machines is achieved through a semantic description of a particular machine. The **code** is compacted using a greedy heuristic strategy. **Optimization** techniques are applied to the intermediate form by shape analysis, during code generation through cost analysis, and during compaction through an optimal **graph** coloring inherent to the compaction strategy. (9 Refs)
Subfile: C
Descriptors: **firmware** ; high level languages; microprogramming; program compilers; software portability
Identifiers: optimisation; portable microcode; retargetable microcode generator system; RMGS; six-stage translation system; C-like HLL; high-level language; horizontal microcode; intermediate semantic description language; ISDL; machine-dependent microcode; heuristic pattern-matched code generator; greedy heuristic strategy; shape analysis; cost analysis; compaction; optimal **graph** coloring
Class Codes: C6150C (Compilers, interpreters and other processors); C5220 (Computer architecture); C6110 (Systems analysis and programming)

31/5/38 (Item 1 from file: 144)
DIALOG(R)File 144:Pascal

16543089 PASCAL No.: 04-0190978
FICO: A fast instruction cache optimizer
SCOPES 2003 : software and compilers for embedded systems : Vienna, 24-26 September 2003
GARATTI Marco
KRALL Andreas, ed
STMicroelectronics, Unknown
International workshop on software compilers for embedded systems, 7 (Vienna AUT) 2003-09-24
Journal: Lecture notes in computer science, 2003, 2826 388-402
ISBN: 3-540-20145-9 ISSN: 0302-9743 Availability: INIST-16343; 354000117799810260
No. of Refs.: 11 ref.
Document Type: P (Serial); C (Conference Proceedings) ; A (Analytic)
Country of Publication: Germany
Language: English
This paper shows the results obtained by FICO, a tool aimed at reducing instruction cache conflict misses. FICO reorders functions without requiring any program execution to gather profiling information. The control flow **graph** annotated with estimated execution frequencies is the actual input of the algorithm. The tool has been implemented as a post linking phase in a newly developed state-of-the-art commercial-quality compiler codesigned by STMicroelectronics and Hewlett-Packard for their embedded processor family LX. Experimental results show that FICO can provide a **speed** -up of about 8% on **embedded applications** .

English Descriptors: Cache memory; Program execution; Data flow; Compiler; Boarded computer; Conflict; Flow control; Fluence **graph** ; Flow **graphs** ; Codesign; **Graph** flow

French Descriptors: Antememoire; Execution programme; Flot donnee; Compilateur; Calculateur embarque; Conflit; Commande ecoulement; **Graphe** fluence; **Graphe** flux; Conception conjointe; Flot **graphe**

Classification Codes: 001D02B01

31/5/43 (Item 4 from file: 34)

DIALOG(R)File 34:SciSearch(R) Cited Ref Sci

02912176 Genuine Article#: MP973 Number of References: 11
Title: PROGRAM IMPLEMENTATION SCHEMES FOR HARDWARE-SOFTWARE SYSTEMS
Author(s): GUPTA RK; COELHO CN; DEMICHELI G
Corporate Source: UNIV ILLINOIS,DEPT COMP SCI,DIGITAL COMP LAB,1304 W SPRINGFIELD AVE,ROOM 2214/URBANA//IL/61801; STANFORD UNIV,DEPT ELECT ENGN/STANFORD//CA/94305
Journal: COMPUTER, 1994, V27, N1 (JAN), P48-55
ISSN: 0018-9162
Language: ENGLISH Document Type: ARTICLE
Geographic Location: USA
Subfile: SciSearch; CC ENGI--Current Contents, Engineering, Technology & Applied Sciences
Journal Subject Category: COMPUTER SCIENCE, HARDWARE & ARCHITECTURE; COMPUTER SCIENCE, SOFTWARE, GRAPHICS, PROGRAMMING
Abstract: Recent advances in large-scale integrated circuit design have prompted system architects to investigate synthesized systems containing both application-specific and general-purpose reprogrammable components. However, software component design for such systems poses interesting problems because serial program execution must interact with concurrent hardware operations.

This article outlines a hardware/software codesign methodology that works to surmount these difficulties. The authors first model system behavior using a hardware description language that supports timing and resource constraints. The HDL description is compiled into a system **graph** model and is then partitioned to satisfy overall system implementation and applicable data-rate constraints.

The software component is implemented as a set of program routines called threads. and the concurrency inherent in the system model is preserved by interleaving the execution of these **threads** .

To illustrate the **effectiveness** of cosynthesis, both cost- and performance-wise, the authors present a hardware/software interface implementation for a graphics controller.

Reprogrammable processors offer a promising means to realize low-cost **embedded applications** . Where possible, portions of system functionality can be delegated to the software component instead of ASICs.
Cited References:
1992 INT WORKSH HARD, 1993
CAMPOSANO R, 1991, HIGH LEVEL VLSI SYNT
CONWAY M, 1963, V6, P396, CACM
DEMICHELI G, 1990, V7, P37, IEEE DES TEST COMPUT
GAJSKI D, 1988, SILICON COMPILATION
GUPTA RK, 1993, COSYNTHESIS HARDWARE
GUPTA RK, 1993, V10, P37, IEEE DESIGN TEST COM
GUPTA RK, 1992, P2, P EUROPEAN DESIGN AU
GUPTA RK, 1992, P225, 29TH P DES AUT C LOS
HENNESSY JL, 1990, P89, COMPUTER ARCHITECTUR
KING PJH, 1967, V10, COMPUTER J

31/5/46 (Item 1 from file: 60)
DIALOG(R)File 60:ANTE: Abstracts in New Tech & Engineer

0000321445 IP ACCESSION NO: AN140742
Design of an optimal folding mechanism for Java processors.

Ton, L.-R.; Chang, L.-C.; Shann, J.-J.; Chung, C.-P.

Microprocessors and Microsystems, v 26, n 8, p p.341-52, 10 Nov. 2002
PUBLICATION DATE: 2002

PUBLISHER: Elsevier Science Ltd., Oxford Fulfillment Centre, P.O. Box 800, Kidlington, Oxford, OX5 1DX
COUNTRY OF PUBLICATION: UK
PUBLISHER URL: http://www.elsevier.com

RECORD TYPE: Abstract
LANGUAGE: English
ISSN: 0141-9331
NOTES: il.; tbls.; refs.
FILE SEGMENT: ANTE: Abstracts in New Technologies and Engineering

ABSTRACT:
Java has become the most important language in the Internet area and is suitable for smart phones, PDAs (personal digital assistants), Internet TVs or other consumer and embedded products. However, the performance of a Java processor is severely limited by the true data **dependency** inherited from the stack architecture defined by Sun's Java Virtual Machine (JVM) that executes the machine **code** known as bytecode. To **enhance** the performance of the JVM, a stack operations folding mechanism for the picoJava-II processor was proposed by Sun Microsystems to fold 42% stack push/pop instructions. A systematic folding algorithm - Producer, Operator, and Consumer (POC) folding model was proposed in earlier research to eliminate up to 82.9% of stack push/pop instructions. The remaining push/pop instructions cannot be folded due to the sequential checking characteristics of the POC folding model. In this paper, a new folding algorithm - enhanced POC (EPOC) - is proposed to further fold the remaining push/pop instructions. Simulation results and statistical data are presented and discussed. (Original abstract - amended)

DESCRIPTORS: Microprocessors; **Embedded** ; **Applications programs** ; Object oriented; Programming; Languages; Java; Compilers; Abstract machines; Intermediate languages; Java byte **code** ; Stack operations; **Enhancement** ; Algorithms; Microprocessors; **Embedded** ; **Applications programs** ; **Object o riente** d; Programming; Languages; Java; Compilers; Abstract machines; Intermediate languages; Java byte **code** ; Stack operations; **Enhancement** ; Algorithms

```
File 347:JAPIO Dec 1976-2006/Dec(Updated 070403)
         (c) 2007 JPO & JAPIO
File 350:Derwent WPIX 1963-2007/UD=200736
         (c) 2007 The Thomson Corporation

Set     Items   Description
S1      73808   DAG OR DIGRAPH OR GRAPH? ? OR DEPENDENCY OR CALL()(DIAGRAM?
              ? OR GRAPH? ? OR CHART? ?) OR TOPOLOGICAL()ORDER???
S2      69361   (CALL??? OR INVOK??? OR INVOCATION)(7N)(PROGRAM? ? OR OBJE-
             CT? ? OR CODE OR CLASS OR CLASSES OR MODULE? ? OR APPLICATION?
              ? OR SOFTWARE OR FIRMWARE OR FILE? ? OR METHOD? ? OR ROUTINE?
              ? OR SUBROUTINE? ? OR THREAD? ? OR PROCESS OR PROCESSES)
S3     148005   (PASS OR PASSES OR PASSING OR PASSAGE OR PASSED OR STAMP???
             OR COUPL??? OR RECEIV???)(5N)(MESSAGE? ? OR ARGUMENT? ? OR P-
             ARAMETER? ? OR VARIABLE? ? OR METADATA OR META()DATA OR SIGNA-
             TURE? ? OR INTERFACE? ?)
S4    9471471   PROGRAM? ? OR OBJECT? ? OR CODE OR CLASS OR CLASSES OR MOD-
             ULE? ? OR APPLICATION? ? OR SOFTWARE OR FIRMWARE OR FILE? ? OR
              METHOD? ? OR ROUTINE? ? OR SUBROUTINE? ? OR THREAD? ?
S5      79563   S4(5N)(OPTIMIZ??? OR OPTIMIS??? OR OPTIMIZATION? ? OR OPTI-
             MISATION? ? OR FINETUN??? OR FINE()TUN??? OR ENHANC??????)
S6      66073   (INCREAS??? OR AUGMENT? OR BOOST? OR AMPLIF? OR RAIS??? OR
             IMPROV??? OR RAMP??? OR UPGRAD?)(5N)S4(5N)(EFFICIENC??? OR EF-
             FICIENT OR EFFECTIV? OR SPEED)
S7       8078   FIRMWARE OR EMBEDDED()(CODE OR PROGRAM? ? OR SOFTWARE OR A-
             PPLICATION? ?)
S8       1584   S1 AND S5:S6
S9          3   S8 AND S7
S10        34   S1 AND S7
S11         7   S2:S3 AND S10
S12        34   S9:S11
S13        24   S1 AND (MICROCODE OR MICRO()CODE OR BIOS)
S14         2   S13 AND S2:S3
S15        36   S12 OR S14
S16        22   S15 AND PY=1963:2003
S17        27   S15 AND AC=US/PR AND AY=(1963:2003)/PR
S18        28   S15 AND AC=US AND AY=1963:2003
S19        28   S15 AND AC=US AND AY=(1963:2003)/PR
S20        29   S16:S19
S21        29   IDPAT (sorted in duplicate/non-duplicate order)
```

21/5,K/1 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0016328187 - Drawing available
WPI ACC NO: 2007-044356/200705
Related WPI Acc No: 2004-478823
XRPX Acc No: N2007-030748
Analyzing method for executable software code involves providing intermediate representation of executable software code **based on** optimized **data flow** graph **and optimized control flow** graph **to facilitate analysis of executable software code**
Patent Assignee: RIOUX C R (RIOU-I)
Inventor: RIOUX C R
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20060253841	A1	20061109	US 2002314005	A	20021206	200705 B
			US 2006415442	A	20060501	

Priority Applications (no., kind, date): US 2002314005 A 20021206; US 2006415442 A 20060501

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20060253841	A1	EN	14	3	Continuation of application US 2002314005
					Continuation of patent US 7051322

Alerting Abstract US A1
NOVELTY - The intermediate representation of an executable software **code** is provided based on an **optimized** data flow **graph** and an optimized control flow **graph** to facilitate analysis of the executable **software code** . The **optimized** exhaustive control flow model and data flow model are provided based on the executable software code.
DESCRIPTION - An INDEPENDENT CLAIM is included for an executable software code analyzing system.
USE - For analyzing executable software code.
ADVANTAGE - Uses nano-code decompiler to determine if flaws, vulnerabilities or general quality issues exist in the code. Allows full representation of control and data flows of target program such that all instructions and internal processes are fully represented at a nano-code level.
DESCRIPTION OF DRAWINGS - The figure shows the block diagram of a nano-code decompilation process.

Title Terms/Index Terms/Additional Words: METHOD; EXECUTE; SOFTWARE; CODE; INTERMEDIATE; REPRESENT; BASED; OPTIMUM; DATA; FLOW; **GRAPH** ; CONTROL; FACILITATE; ANALYSE

Class Codes
International Classification (+ Attributes)
IPC + Level Value Position Status Version
G06F-0009/44 A I F B 20060101
US Classification, Issued: 717127000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-J10C1; T01-J20B

21/5,K/5 (Item 5 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0014692220 - Drawing available
WPI ACC NO: 2005-039809/200504
XRPX Acc No: N2005-034687
Test case generation method used for testing hardware application, involves providing specification of application in form of flow graph **containing loop, by user, and generating test cases based on flow** graph
Patent Assignee: DECCANET DESIGNS LTD (DECC-N)

Inventor: REGUNATHAN V
Patent Family (1 patents, 106 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
WO 2004107087	A2	20041209	WO 2004IN145	A	20040528	200504 B

Priority Applications (no., kind, date): US 2003473883 P 20030529

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
WO 2004107087	A2	EN	37	9	

National Designated States,Original: AE AG AL AM AT AU AZ BA BB BG BR BW BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NA NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW
Regional Designated States,Original: AT BE BG BW CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NA NL OA PL PT RO SD SE SI SK SL SZ TR TZ UG ZM ZW

Alerting Abstract WO A2

NOVELTY - The specification of application, is provided in the form of flow **graph** containing loop, by a user using a computer system. The test cases for testing the application, is generated based on the flow **graph** .

DESCRIPTION - An INDEPENDENT CLAIM is also included for recorded medium storing test case generation program.

USE - For generating test cases used for testing hardware, software and **firmware** applications.

ADVANTAGE - By generating test cases based on flow **graph** containing nodes, the application is effectively tested.

DESCRIPTION OF DRAWINGS - The figure shows the flowchart explaining test case generation process.

Title Terms/Index Terms/Additional Words: TEST; CASE; GENERATE; METHOD; HARDWARE; APPLY; SPECIFICATION; FORM; FLOW; **GRAPH** ; CONTAIN; LOOP; USER; BASED

Class Codes
International Classification (Main): G06F

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-J20C; T01-S03

...method used for testing hardware application, involves providing specification of application in form of flow graph **containing loop, by user, and generating test cases based on flow** graph

Alerting Abstract ...NOVELTY - The specification of application, is provided in the form of flow **graph** containing loop, by a user using a computer system. The test cases for testing the application, is generated based on the flow **graph** . ...USE - For generating test cases used for testing hardware, software and **firmware** applications...

...ADVANTAGE - By generating test cases based on flow **graph** containing nodes, the application is effectively tested...

Title Terms.../Index Terms/Additional Words: **GRAPH** ;

Original Publication Data by Authority

Original Abstracts:
A user may provide the specification of an application in the form of flow-**graphs** to a computer **system** . The computer system examines (traverses) the flow **graphs** to generate various **test** paths. The user may then specify different input value combinations associated with each test path ...

21/5,K/8 (Item 8 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0013918132 - Drawing available
WPI ACC NO: 2004-097890/ **200410**
XRPX Acc No: N2004-077988

Firmware **component installation method for flash memory, involves loading** firmware **component upgrade into** firmware **memory, when** firmware **component upgrade is compatible with pre-stored** firmware **component**

Patent Assignee: BOLDON J L (BOLD-I); LEE L C (LEEL-I)
Inventor: BOLDON J L; LEE L C

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 20030233493	A1	20031218	US 2002173326	A	20020615	200410 B

Priority Applications (no., kind, date): US 2002173326 A 20020615

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 20030233493	A1	EN	15	6	

Alerting Abstract US A1

NOVELTY - The **firmware** components and **firmware** component dependent data are stored in a memory. The **firmware** dependent data of the **firmware** component upgrade to be loaded into the memory is read. The **firmware** upgrade component is determined to be compatible with the stored **firmware** components, based on the read data, to load the **firmware** upgrade into the memory.

DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

1. firmware bundle; and

2. firmware supporting device.

USE - For installing **firmware** component into **firmware** supporting device **e** .g. industrial **equipment** , vehicle, telecommunication device, digital camera and memory device e.g. erasable programmable read only memory (EPROM), one time programmable read only memory (OTPROM), electrically erasable programmable read only memory (EEPROM) and flash memory

ADVANTAGE - Prevents potentially the installation of incompatible **firmware** components into **firmware** supporting device.

DESCRIPTION OF DRAWINGS - The figure shows the schematic diagram of **firmware** component apparatus and the **firmware bundle** .

Title Terms/Index Terms/Additional Words: **FIRMWARE** ; COMPONENT; INSTALLATION; METHOD; FLASH; MEMORY; LOAD; UPGRADING; COMPATIBLE; PRE; STORAGE

Class Codes
International Classification (Main): G06F-003/00
US Classification, Issued: 710001000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F01B1; T01-F05B2; T01-H01B3; T01-S03

Firmware **component installation method for flash memory, involves loading** firmware **component upgrade into** firmware **memory, when** firmware **component upgrade is compatible with pre-stored** firmware **component**

Original Abstracts:

Methods and apparatus to facilitate prevention of the installation of incompatible **firmware** components into a **firmware** -supported device. Methods **include** performing an automatic **firmware dependency** compatibility check **in conjunction** with an attempted installation of **firmware** components into the **firmware** -supported device. Apparatus **include** a **firmware** bundle having one **or** more **firmware** components and which **also** has a **firmware** compatibility identifier that **is** configured to facilitate the performance of the automatic **firmware dependency** compatibility check. **Other apparatus** include a **firmware** -supported device that **comprises** a **firmware** component **dependency** compatibility

algorithm that **is** configured to prevent the installation of incompatible **firmware** components into the **firmware** -supported device.

Claims:
What is claimed is: 1. A method, comprising automatically checking **firmware** component **dependency** compatibility in conjunction with installation of **firmware** components **into** a **firmware** -supported device.

21/5,K/12 (Item 12 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0013328867 - Drawing available
WPI ACC NO: 2003-416239/ **200339**
XRPX Acc No: N2003-331748
Microprocessor for computer system, has reservation station including checking circuitry for checking dependency **of carry flag or overflow flag**
Patent Assignee: ADVANCED MICRO DEVICES INC (ADMI)
Inventor: CLARK M T; MAHURIN E W
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 6535972	B1	20030318	US 1999441631	A	19991116	200339 B

Priority Applications (no., kind, date): US 1999441631 A 19991116

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6535972	B1	EN	16	7	

Alerting Abstract US B1
NOVELTY - The microprocessor uses either the carry flag/overflow flag (304) as source operands (301,302) based on instruction (300) received by a reservation station. The reservation station includes a checking circuitry for checking the **dependency** of the carry flag or overflow flag.
DESCRIPTION - An INDEPENDENT CLAIM is also included for microprocessor operation method.
USE - Microprocessor e.g. x86 processor for computer system.
ADVANTAGE - The shared **dependency** checking of the status flags reduces the amount of circuitry present in the reservation station, thereby saving chip area.
DESCRIPTION OF DRAWINGS - The figure illustrates a storage location in the reservation station.
300 instruction
301,302 source operands
304 carry flag/overflow flag

Microprocessor for computer system, has reservation station including checking circuitry for checking dependency **of carry flag or overflow flag**

Original Titles:
Shared **dependency** checking for status flags

Alerting Abstract ...received by a reservation station. The reservation station includes a checking circuitry for checking the **dependency** of the carry flag or overflow flag....ADVANTAGE - The shared **dependency** checking of the status flags reduces the amount of circuitry present in the reservation station...

Original Publication Data by Authority

Original Abstracts:
A system and method for shared **dependency** checking of status **flags** . In certain instruction set architectures, the reading of some status flags by the instruction set...

...the reading of other status flags. Hardware for exclusively read flags may be shared in **dependency** checking circuitry, allowing **the** reading of either one flag or the other for during **dependency** checking of a **given** instruction. This may allow circuit area to be saved. For an instruction

that may need access to both flags, system **firmware** (e.g. microcode) **can** be used to break the instruction into two separate instructions or operations, thereby allowing the...

21/5,K/14 (Item 14 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2007 The Thomson Corporation. All rts. reserv.

0013119991 - Drawing available
WPI ACC NO: 2003-201783/ **200319**
XRPX Acc No: N2003-160721
Executable radio software system using embedded processors has isolated platform dependent code in one or more files for number of different platforms each selectively compilable by a directive
Patent Assignee: BICKLE G L (BICK-I); EYERMANN P A (EYER-I); MARKS J T (MARK-I); RAYTHEON CO (RAYT)
Inventor: BICKLE G L; EYERMANN P A; MARKS J T
Patent Family (5 patents, 99 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update	
WO 2003012638	A1	20030213	WO 2002US24042	A	20020729	200319	B
US 20030114163	A1	20030619	US 2001308270	P	20010727	200341	E
			US 2002207315	A	20020729		
EP 1412851	A1	20040428	EP 2002748267	A	20020729	200429	E
			WO 2002US24042	A	20020729		
AU 2002317596	A1	20030217	AU 2002317596	A	20020729	200452	E
JP 2004537803	W	20041216	WO 2002US24042	A	20020729	200482	E
			JP 2003517748	A	20020729		

Priority Applications (no., kind, date): US 2002207315 A 20020729; US 2001308270 P 20010727

Alerting Abstract WO A1
NOVELTY - The system includes a core framework layer responsive to one or more applications and a middleware layer. The core framework layer includes an isolated platform dependent code in one or more files for a number of different platforms each selectively compilable by a directive to reduce the **dependency** of the core framework layer on a specific platform. The middleware layer includes CORBA and RTOS.
DESCRIPTION - An INDEPENDENT CLAIM is included for an executable radio software method
USE - As software communication architecture for radios such as handheld with multiple bands and other applications using embedded microprocessors.
ADVANTAGE - Platform independent, (i.e., operates on multiple embedded processors and with multiple RTOS and ORBs) and is written such that a compile option allows the software to do so. Uses a distributed XML parser to parse application XML files. Enables the user to install and uninstall software in a SCA compliant radio system and to enable the user to launch and tear down SCA compliant radio applications. Performs the parsing function only when a new application or device is installed. Prevents an application from attempting to address a device when it is busy, off-line, or disabled. Restarts a device or application upon failure and also may notify the user that a failure has occurred.
DESCRIPTION OF DRAWINGS - The drawing shows the core framework layer IDL relationships.

Title Terms/Index Terms/Additional Words: EXECUTE; RADIO; SOFTWARE; SYSTEM; EMBED; PROCESSOR; ISOLATE; PLATFORM; DEPEND; CODE; ONE; MORE; FILE; NUMBER; SELECT; DIRECT

Class Codes
International Classification (Main): G06F-009/44, H04Q-007/20
US Classification, Issued: 455450000, 455418000

File Segment: EPI;
DWPI Class: T01; W01
Manual Codes (EPI/S-X): T01-F05A; T01-J08C; T01-N03B2; W01-C01D3C; W01-C01D3J; W01-C01Q3

...for a number of different platforms each selectively compilable by a directive to reduce the **dependency** of the core framework layer on a specific platform. The middleware layer includes CORBA and...

Original Publication Data by Authority

Original Abstracts:
...for a number of different platforms each selectively compilable by a directive to reduce the **dependency** of the core **framework** layer on a specific platform. Also, the core framework layer includes an embedded distributed parser...

21/5,K/17 (Item 17 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0010892278
WPI ACC NO: 2001-512603/ **200156**
XRPX Acc No: N2001-379493
Improved method for starting an embedded application
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
RD 441129	A	20010110	RD 2000441129	A	20001220	200156 B

Priority Applications (no., kind, date): RD 2000441129 A 20001220

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
RD 441129	A	EN	1	0	

Alerting Abstract RD A
NOVELTY - Described is an improved method by which an application can start an **embedded application** like a Java Virtual Machine (JVM). The improved method reduces hard-coded dependencies between the application and the JVM, thus enabling applications to exploit new function sooner without code changes, and making maintenance and tuning more efficient. This disclosure addresses two sources of dependencies between applications and the JVM that can be improved.
DESCRIPTION - This disclosure proposes a way to resolve the two dependencies. First, applications have many dependencies on the internal structure of the JVM they want to start, which makes startup code more complicated and puts restrictions on the way JVMs can be structured to suit various configurations. For example, currently the application must set up a path variable for two directories, a classpath variable for another, and will have a hard-coded **dependency** on the name of the dll containing the startup api. Any of the paths or names can be changed from release to release, for example, because a new debug version is developed, or because two different versions are required due to hardware or operating system dependencies. While the hard-coded **dependency** cannot be completely eliminated without some convention on the part of the application, the number of dependencies can be reduced and the problem of changeability can be eliminated by putting a startup api (a new one in the case of the JVM) in a dll separate from parts of the JVM that may have dependencies, guaranteeing the name of the new dll across releases, and including version information in the JVM startup parameters, so that the startup api can load the appropriate version of the JVM at runtime . In addition the structural **dependency** can be further reduced if applications by convention use an environment variable, or some other external configuration method, to provide the path to find the jvm, and possibly a second environment variable for the name of the dll containing the startup api.
USE - Method for starting an **embedded application** .

Title Terms/Index Terms/Additional Words: IMPROVE; METHOD; START; EMBED; APPLY

Class Codes
International Classification (Main): G06F

21/5,K/19 (Item 19 from file: 350)
DIALOG(R)File 350:Derwent WPIX

0010741189 - Drawing available
WPI ACC NO: 2001-353752/ **200137**
XRPX Acc No: N2001-256896
Complete failure reporting method **for computer system, involves** invoking firmware **mechanism of computer system to reboot the system**
Patent Assignee: SIEMENS INFORMATION & COMMUNICATIONS NET (SIEI)
Inventor: DORWIN P A; SHAPIRO R
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 6230286	B1	20010508	US 1993983719	A	19930128	200137 B
			US 1995408127	A	19950321	

Priority Applications (no., kind, date): US 1993983719 A 19930128; US 1995408127 A 19950321

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6230286	B1	EN	7	3	Continuation of application US 1993983719

Alerting Abstract US B1

NOVELTY - A computer operating system which determines that non-recoverable system error currently exists, **invokes** a **firmware** mechanism of a computer system to reboot the system. The **firmware** mechanism sends a remote location notification of system failure by establishing a data connection with remote site when the system cannot be rebooted.

DESCRIPTION - An INDEPENDENT CLAIM is also included for error reporting method.

USE - For computer system maintenance and diagnostics and for computer system failure reporting mechanism for reporting system failure to remote site.

ADVANTAGE - The **firmware** mechanism reports failure of computer system to remote site without requiring the system to be successfully rebooted.

DESCRIPTION OF DRAWINGS - The figure shows a flow chart which explains interaction between the remote reporting facility and remote host system.

Title Terms/Index Terms/Additional Words: COMPLETE; FAIL; REPORT; METHOD; COMPUTER; SYSTEM; INVOKE; **FIRMWARE** ; MECHANISM

Class Codes
International Classification (Main): G06F-011/14
US Classification, Issued: 714023000

File Segment: EPI;
DWPI Class: T01; U21
Manual Codes (EPI/S-X): T01-G03; U21-A06

Complete failure reporting method **for computer system, involves** invoking firmware **mechanism of computer system to reboot the system**

Alerting Abstract ...NOVELTY - A computer operating system which determines that non-recoverable system error currently exists, **invokes** a **firmware** mechanism of a computer system to reboot the system. The **firmware** mechanism sends a remote location notification of system failure by establishing a data connection with...
...ADVANTAGE - The **firmware** mechanism reports failure of computer system to remote site without requiring the system to be...

Title Terms.../Index Terms/Additional Words: **FIRMWARE** ;

Original Publication Data by Authority

Original Abstracts:
...failed computer system to send a report of the failure to a remote site without **dependency** on a service processor or maintenance processor. The computer system is capable of reporting system...

...a main memory, a storage device storing a computer operating system, and

a data modem. **Firmware** is resident in the non-volatile control store of the main computer system for reporting...

...the system may be successfully rebooted by reloading the operating system into main memory. The **firmware** is **invoked** upon occurrence of a non-recoverable error and provides for assembling failure-related information, establishing...

Claims:

...computer operating system determining that a non-recoverable system error currently exists; the operating system **invoking** a **firmware** mechanism within the **computer** system; **the firmware** mechanism attempting to reboot said system; and **the firmware** mechanism, if the system cannot be rebooted, sending to **a** remote location notification of system failure.

21/5,K/22 (Item 22 from file: 350)

0009824340 - Drawing available
WPI ACC NO: 2000-115368/200010
XRPX Acc No: N2000-087252

Computer-implemented test sub-sequence graph creating method

Patent Assignee: MOTOROLA INC (MOTI)
Inventor: HULL C A; SUN X

Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 6004027	A	19991221	US 1995399020	A	19950306	200010 B

Priority Applications (no., kind, date): US 1995399020 A 19950306

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 6004027	A	EN	40	21	

Alerting Abstract US A

NOVELTY - One test sub-sequence (TS) is constructed for each member of the unique input-output (UIO) sequence set for each state transition. The UIO set is selected for each of the edges under test (EUT). Several test sub-sequence graph vertices corresponding to model states are connected to edges of the TS graph. The edges of the TS graph connect first state of test sub-sequence to the last state.

DESCRIPTION - An UIO set is identified for every model state. Every model state is traversed by a sequence of state transitions (TR) corresponding to each of the member I/O sequence in the UIO sets uniquely identifying the model state.

An INDEPENDENT CLAIM is also included for computer-implemented TS graph creating apparatus.

USE - For testing hardware, firmware, implementation of communication software and VLSI circuits.

ADVANTAGE - A change to one lower level sub-graph or model need not require reconstruction of TSs for the remainder of FSM model, thus hierarchical TS construction can be done easily.

DESCRIPTION OF DRAWINGS - The figure shows flow chart of TS graph creating method.

Title Terms/Index Terms/Additional Words: COMPUTER; IMPLEMENT; TEST; SUB; SEQUENCE; GRAPH; METHOD

Class Codes
International Classification (Main): G06F-011/00
(Additional/Secondary): G01R-031/28
US Classification, Issued: 371027400, 364578000

File Segment: EPI;
DWPI Class: S01; T01; U11
Manual Codes (EPI/S-X): S01-G01A; S01-G02B; T01-G; T01-G02A2D; U11-F01C3; U11-F01G; U11-G

Original Publication Data by Authority

Claims:
...and ends at a Test Subsequence (TS) Last State; and (d) constructing the Test Subsequence **Graph** (TSgraph) for storage in the Memory by connecting a plurality of Test Subsequence **Graph** (TSgraph) Vertices corresponding to Model States to a plurality of Test Subsequence **Graph** (TSgraph) Edges, wherein: each of the plurality of Test Subsequence **Graph** (TSgraph) Vertices corresponds to a different Model State, and each of the plurality of Test Subsequence **Graph** (TSgraph) Edges is a Test Subsequence **Graph** (TSgraph) Edge which connects the corresponding Test Subsequence First State to the Test **Subsequence** Last State.

21/5,K/24 (Item 24 from file: 350)

0008916538 - Drawing available
WPI ACC NO: 1998-467004/ **199840**
XRPX ACC NO: N1998-363835
Euler touring of augmented graph **for testing hardware,** firmware , **software implementations - involves constructing augmented** graph **as micro-edge bridging sequences between corresponding matched augmented** graph **vertices**
Patent Assignee: MOTOROLA INC (MOTI)
Inventor: HULL C A; SUN X
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 5796752	A	19980818	US 1995399392	A	19950306	199840 B

Priority Applications (no., kind, date): US 1995399392 A 19950306

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 5796752	A	EN	62	21	

Alerting Abstract US A
The method uses an FSM model and involves providing control signals to direct a processor to count the number of incoming test subsequence (TS) **graph** micro edges to each TS **graph** vertex and also the number of outgoing TS **graph** micro edges from each TS **graph** vertex. The incoming and outgoing micro edges are the indegree and outdegree for those respective TS **graph** vertices. An augmented **graph** is constructed from the TS **graph** by building micro-edge bridging sequences between TS **graph** vertices with indegree greater than outdegree, and TS **graph** vertices and outdegree greater than indegree.
The TS **graph** vertices are initially sorted based on differences in indegree and outdegree. The TS **graph** vertices with highest difference on subtracting outdegree from indegree are matched with those with highest difference on subtracting indegree from outdegree. The micro edge bridging sequence between corresponding matched augmented **graph** vertices is then built to produce augmented **graph** micro edge. The vertices in the augmented **graph** each have an equal number of incoming and outgoing augmented **graph** micro-edges. The verification test sequence is created by touring the augmented **graph** micro edges.
ADVANTAGE - Detects any practical defects and solves controllability problem.

Title Terms/Index Terms/Additional Words: EULER; TOURING; AUGMENT; **GRAPH** ; TEST; HARDWARE; **FIRMWARE** ; SOFTWARE; CONSTRUCTION; MICRO; EDGE; BRIDGE; SEQUENCE; CORRESPOND; MATCH; VERTEX; FINITE; STATE; MACHINE

Class Codes
International Classification (Main): G06F-011/00
US Classification, Issued: 371027100, 364488000

File Segment: EPI;
DWPI Class: T01
Manual Codes (EPI/S-X): T01-F05; T01-G07A

Euler touring of augmented graph **for testing hardware,** firmware **, software implementations...**

...involves constructing augmented graph **as micro-edge bridging sequences between corresponding matched augmented** graph **vertices**

... **FIRMWARE** ;

Original Publication Data by Authority

Original Abstracts:
...Finite State Machine (33) (FSM) model. The number of incoming and outgoing Test Subsequence (TS) **graph** (39) micro-edges **are** determined for each TS **graph** (39) vertex or **Finite** State Machine (33) state. An Augmented **Graph** (95) is created (**40**) by constructing Test Subsequence (TS) micro-edge bridging sequences between TS **graph** vertices with relatively **more** incoming micro-edges and vertices with relatively more outgoing micro-edges. The newly symmetric Augmented **Graph** (95) is Euler **Toured** (42), generating Verification Test Sequences (43), used to test a Machine-Under-Test (14) for...

21/5,K/26 (Item 26 from file: 350)

0008178091 - Drawing available
WPI ACC NO: 1997-280565/ **199725**
XRPX Acc No: N1997-232507
Hierarchical test subsequence and finite state machine model graph **merging method for constructing verification test sequence - involves merging All Child Test Subsequence Subgraphs with their corresponding Parent Test Subsequence Subgraphs by connecting all incoming and Outgoing Child Test Sub sequence Subgraph Micro-Edges to Test Subsequence Subgraph Vertices**
Patent Assignee: MOTOROLA INC (MOTI)
Inventor: HULL C A; SUN X
Patent Family (1 patents, 1 countries)

Patent Number	Kind	Date	Application Number	Kind	Date	Update
US 5630051	A	19970513	US 1995399307	A	19950306	199725 B

Priority Applications (no., kind, date): US 1995399307 A 19950306

Patent Details

Number	Kind	Lan	Pg	Dwg	Filing Notes
US 5630051	A	EN	57	21	

Alerting Abstract US A
The method involves identifying the Parent Test Subsequence Sub **graph** for each Child Test Subsequence Subgraph. All incoming Test Subsequence Sub **graph** Micro-Edges and all Outgoing Test Subsequence Subgraph Micro-Edges for Each Child Test Sub sequence Subgraph are identified. All Child Test Subsequence Subgraphs are merged with their corresponding Parent Test Subsequence Subgraphs by connecting all incoming Child Test Sub sequence Subgraph Micro-Edges and all Outgoing Child Test Subsequence Subgraph Micro-Edges to Test Subsequence Subgraph Vertices. The Merged Test Subsequence **Graph** for storage in the memory are forming by combining all of the one or more Merged Parentless Test Subsequence Sub **graphs** .
ADVANTAGE - Provides simple hardware, software and **firmware** for testing electronics circuits.

Non-volatile memory

From Wikipedia, the free encyclopedia

Non-volatile memory, **nonvolatile memory**, **NVM** or **non-volatile storage**, is computer memory that can retain the stored information even when not powered. Examples of non-volatile memory include read-only memory, flash memory, most types of magnetic computer storage devices (e.g. hard disks, floppy disk drives, and magnetic tape), optical disc drives, and early computer storage methods such as paper tape and punch cards.

Non-volatile memory is typically used for the task of secondary storage, or long-term persistent storage. The most widely used form of primary storage today is a volatile form of random access memory (RAM), meaning that when the computer is shut down, anything contained in RAM is lost. Unfortunately, most forms of non-volatile memory have limitations that make them unsuitable for use as primary storage. Typically, non-volatile memory either costs more or performs worse than volatile random access memory.

Several companies are working on developing non-volatile memory systems comparable in speed and capacity to volatile RAM. For instance, IBM is currently developing MRAM (Magnetic RAM). Not only would such technology save energy, but it would allow for computers that could be turned on and off almost instantly, bypassing the slow start-up and shutdown sequence.

Computer memory types

Volatile

- DRAM, e.g. DDR SDRAM
- SRAM
- Upcoming
  - Z-RAM
  - TTRAM
- Historical
  - Williams tube
  - Delay line memory

Non-Volatile

- ROM
  - PROM
  - EAROM
  - EPROM
  - EEPROM
  - Flash memory
- Upcoming
  - FeRAM
  - MRAM
  - PRAM
  - SONOS
  - RRAM
  - NRAM
- Historical
  - Drum memory
  - Magnetic core memory
  - Bubble memory

Non-volatile data storage can be categorised in electrically addressed systems random access memory and mechanically addressed systems hard disks, optical disc, magnetic tape, Holographic memory and such. Electrically addressed systems are expensive, but fast, whereas mechanically addressed systems have a low price per bit, but are slow. Non-volatile memory may one day eliminate the need for comparatively slow forms of secondary storage systems, which include hard disks.

Contents

Electrically addressed

Electrically addressed non-volatile memories based on charge storage can be categorised according to their write mechanism:

Mask-programmed ROM

One of the earliest forms of non-volatile read-only memory, the mask-programmed ROM was prewired at the design stage to contain specific data; once the mask was used to manufacture the integrated circuits, the data was cast in stone (silicon, actually) and could not be changed.

The mask ROM was therefore useful only for large-volume production, such as for read-only memories containing the startup code in early microcomputers. This program was often referred to as the "bootstrap", as in pulling oneself up by one's own bootstraps.

Due to the very high initial cost and inability to make revisions, the mask ROM is rarely if ever used in new designs.

Programmable ROM

See main article Programmable read-only memory.

The next approach was to create a chip which was initially blank; the programmable ROM originally contained silicon or metal fuses, which would be selectively "blown" or destroyed by a device programmer or PROM programmer in order to change 0s to 1s. Once the bits were changed, there was no way to restore them to their original condition. Non-volatile but still somewhat inflexible.

Early PAL programmable array logic chips used a similar programming approach to that used in the fuse-based PROMs.

In most new designs, eraseable memories or one-time programmable chips have replaced the old fuse PROMs.

Erasable PROMs

See main article EPROM.

There are two classes of non-volatile memory chips based on EPROM technology.

UV-erase EPROM

The original erasable non-volatile memories were EPROM's; these could be readily identified by the distinctive quartz window in the centre of the chip package. These operated by trapping an electrical charge on the gate of a field-effect transistor in order to change a 1 to a 0 in memory. To remove the charge, one would place the chip under an intense short-wavelength fluorescent ultraviolet lamp for 20-30 minutes, returning the entire chip to its original blank (all ones) state.

OTP (one-time programmable) ROM

An OTP is electrically an EPROM, but with the quartz window physically missing. Like the fuse PROM it can be written once, but cannot be erased. It has largely replaced PROM chips in electronic production as an EPROM with no window is inexpensive to manufacture and can be programmed using identical equipment to that used to write to the UV-window EPROM.

Electrically erasable PROM

See main article EEPROM.

Electrically erasable PROM's have the advantage of being able to selectively erase any part of the chip without the need to erase the entire chip and without the need to remove the chip from the circuit. While an erase and rewrite of a location appears nearly instantaneous to the user, the write process is slightly slower than the read process; the chip can be read at full system speeds.

The limited number of times a single location can be rewritten is usually in the 10000-100000 range; the capacity of an EEPROM also tends to be smaller than that of other non-volatile memories. Nonetheless, EEPROMs are useful for storing settings or configuration for devices ranging from dial-up modems to satellite receivers.

Flash memory

See main article Flash memory.

The flash memory chip is a close relative to the EEPROM; it differs in that it can only be erased one block or "page" at a time. Capacity is substantially larger than that of an EEPROM, making these chips a popular choice for digital cameras and desktop PC BIOS chips.

Battery-backed static RAM

See main article Nonvolatile BIOS memory.

This is a volatile memory chip (which loses its data if power is removed) to which a battery has been added in order to preserve the contents in the absence of external power. These used to be typically manufactured with CMOS technology to minimise power consumption; a lithium cell can easily power a small memory for a few years. It is now common to use SDRAM with a Lithium ion battery; it is possible to preserve a gigabyte of such memory for days. The settings from the BIOS menus which appear on startup on most desktop PCs are stored in battery-backed CMOS static RAM as a battery must

already be present on the mainboard to keep the real-time clock running when the computer is not in use.

An example of this is seen in video game consoles that allow you to save your games onto the cartridges, such as the Nintendo 64 or the Game Boy.

non volatile Static RAM (nvSRAM)

See main article nvSRAM.

See also

- NVRAM

Mechanically addressed systems

Tape

See main article Magnetic tape.

Hard disk

See main article Hard disk.

Optical disk

See main article Optical disc.

Nanodrive

See main article IBM Millipede.

Holographic storage

See main article Holographic memory.

Specifications

Include table with specifications (Data density, capacity, price per bit, price per unit, data rate, access time, power consumption, form factor) of Flash, Hard disc (2.5", 1"), Tape, DVD, MRAM, NanoDrive and Holographic Disc . Table will be kept up to date on a yearly basis

A justification for this table can be found here (http://www.bluwiki.com/go/Informationstoragecourse):

Retrieved from "http://en.wikipedia.org/wiki/Non-volatile_memory"

Categories: Articles to be merged since February 2007 | Computer memory | Computer storage | Non-

volatile memory